TE
662
.A3
no.
FHWA-
RD-
76-67

Report No. FHWA-RD-76-67

VEHICLE DETECTION--PHASE III
PASSIVE BUS DETECTOR / INTERSECTION PRIORITY SYSTEM DEVELOPMENT

Vol. II, Part I.
Appendixes A, B, and C--Computer Program Documentation

Dept. of Transportation
MAR 10 1977
Library

December 1975
Final Report

This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

Prepared for
FEDERAL HIGHWAY ADMINISTRATION
Offices of Research & Development
Washington, D.C. 20590

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The contents of this report reflect the views of the contracting organization, which is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policy of the Department of Transportation. This report does not constitute a standard, specification, or regulation.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein only because they are considered essential to the object of this document.

TC
662
.A3
no.
FHWA-
RD-
76-67

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-RD- 76-67		

4. Title and Subtitle	5. Report Date
VEHICLE DETECTION - PHASE III PASSIVE BUS DETECTOR/INTERSECTION PRIORITY SYSTEM DEVELOPMENT Volume II, Part I: Appendices A, B, and C*	December 1975
	6. Performing Organization Code
7. Author(s) P. Anderson, R. Larson, R. Lubke, G. Putnam, and D. Wick	8. Performing Organization Report No. 2551-40274
9. Performing Organization Name and Address Honeywell Inc. 2600 Ridgway Parkway Minneapolis, MN. 55413	10. Work Unit No. (TRAIS) FCP-32L1012
	11. Contract or Grant No. DOT-FH-11-8149
	13. Type of Report and Period Covered Final report
12. Sponsoring Agency Name and Address Federal Highway Administration Office of Research and Development Washington, D.C. 20590	14. Sponsoring Agency Code T-0182

15. Supplementary Notes

*Computer Program Documentation

16. Abstract

This volume contains the description of the software developed for the Passive Bus Detector/Intersection Priority System. The program documentation is formatted according to the Program Documentation section of the FHWA Documentation Standards. The computer program documentation is referenced in other volumes of this report as the following appendices:

Appendix A. FORTRAN IV Version of the Vehicle Classification Algorithm
Appendix B. BPCU Program for Minneapolis Installation
Appendix C. BPCU Program for Washington Installation

Appendix C is to be treated as an addendum to Appendix B and contains documentation of only those modules which are different from the Minneapolis Installation program in Appendix B.

Appendix D, Software and Computer Printouts, is presented separately as Part II of this volume.

Dept. of Transportation
MAR 10 1977
Library

17. Key Words	18. Distribution Statement
Passive bus detector, ILD signal processing, Microprocessor, Fully independent controller	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	320	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

PREFACE

The software developed for the passive bus detector system is documented in Appendices A, B, C, and D which are contained in Volume II, Parts 1 and 2.

APPENDIX A. FORTRAN IV VERSION OF THE VEHICLE CLASSIFICATION ALGORITHM

This program duplicates the logic and arithmetic operations of the vehicle classification algorithm that operates in the microcomputer unit of the Bus Preempt Control Unit (BPCU). This program is written in FORTRAN IV for operation on a general-purpose computer. Using this program, the vehicle classification can be analyzed off line with respect to the classification or rejection of particular vehicle signatures.

APPENDIX B. BUS PREEMPT CONTROL UNIT PROGRAM FOR MINNEAPOLIS INSTALLATION

This program is written for the field test of the Passive Bus Detector system in Minneapolis. The detector configuration consists of a pair of detectors (upstream preempt, downstream clearance) on opposing approaches along the main street; no detectors are placed on the cross street. The preempt operation includes both green extension on the main street and green advance by contraction of the cross street green. The timing plan for the traffic signals consists of two sequences of intervals. Traffic flows both directions on both streets. Opposite directions are controlled identically. There is no special signalization for the buses or for turning movements.

APPENDIX C. BUS PREEMPT CONTROL UNIT PROGRAM FOR WASHINGTON INSTALLATION

This program is written for the field test of the Passive Bus Detector system in Washington, D. C. The detector configuration consists of one detector in a curb-side, reserved bus lane on the main street; the cross street access from one side makes a T intersection. The preempt operation involves the insertion of a sequence of intervals on the special bus signalization and stopping of the parallel, same-way traffic on the main street. The bus is detected and classified downstream of the passenger loading zone and upstream of the stop bar. The exclusive green for the bus may be extended incrementally for each additional bus during the timing of that interval. A bus classification during the cross street sequence causes the insertion of the bus sequence in the next cycle.

TABLE OF CONTENTS

		Page
Appendix A –	FORTRAN IV Version of the Vehicle Classification Algorithm	1
3.0	Program Documentation	1
3.1	Main Program	2
3.2	Subroutine VCLAS	87
3.3	Subroutine ERE	96
Appendix B –	Bus Preempt Control Unit Program for Minneapolis Installation	99
3.0	Program Documentation	99
3.1	Assembly Constants	103
3.2	Base Page/Data Base CPC	104
3.3	Executive CPC	120
3.4	Data Input CPC	200
3.5	Data Output CPC	206
3.6	Timing Plan Execution CPC	218
3.7	Bus Preempt CPC	239
3.8	Program Listings and Flowcharts	279
Appendix C –	Bus Preempt Control Unit Program for Washington Installation	292
3.0	Program Documentation	292
3.2.2.4	Preempt Data Base	294
3.6.6	Timing Plan Data Tables	298
3.7	Bus Preempt CPC	300
3.8	Program Listings and Flowcharts	307

iv

LIST OF ILLUSTRATIONS

Figure		Page
A-1	Flowchart of the FORTRAN IV Vehicle Classifier Main Program	9
A-2	Flowchart of the FORTRAN IV Vehicle Classifier Vehicle Classification Routine (VCLAS)	90
A-3	Flowchart of the FORTRAN IV Vehicle Classifier Error Handling Routine	98
B-1	Memory Map of MCU	100
B-2	Passive Bus Detector/Bus Preempt Software Operation	102
B-3	Restart Initialization Routine (Executive CPC)	123
B-4	General Interrupt Servicing Routine (Executive CPC)	127
B-5	Background Program Monitor Routine (Executive CPC)	131
B-6	Clock Error Routine (Executive CPC)	133
B-7	Scheduling Routine (Executive CPC)	136
B-8	Disconnect Monitor Routine (Executive CPC)	138
B-9	Command Monitoring Routine (Executive CPC)	141
B-10	Synchronization Routine (Executive CPC)	144
B-11	Interconnect Data Monitor Routine (Executive CPC)	147
B-12	Timing Routine (Executive CPC)	149
B-13	End of Cycle Routine (Executive CPC)	152
B-14	Status Display Routine (Executive CPC)	156
B-15	Detector Monitoring Routine (Executive CPC)	159
B-16	Autodisconnect Routine (Executive CPC)	162
B-17	Programming Panel Initialization Routine (Executive CPC)	165

LIST OF ILLUSTRATIONS (Continued)

Figure		Page
B-18	Programming Panel Real Time Monitor Routine (Executive CPC)	169
B-19	Process Programming Panel Plan 1 Data Routine (Executive CPC)	174
B-20	Process Programming Panel Plan 1 Data Routine (Executive CPC)	175
B-21	Process Programming Panel Plan 3 Data Routine (Executive CPC)	176
B-22	Update Programming Panel Data Routine (Executive CPC)	178
B-23	Convert Binary Coded Digits to Integer Routine (Executive CPC)	181
B-24	Routine to Multiply an Integer by 10 (Executive CPC)	184
B-25	Routine to Multiply an Integer by 24 (Executive CPC)	185
B-26	Error Recording Routines (Executive CPC)	187
B-27	Initialize Computer Control Unit Routine (Executive CPC)	190
B-28	Initialization RAM Address Test Monitor Routine (Executive CPC)	193
B-29	RAM Address Test Routine (Executive CPC)	196
B-30	Catch-on-the-Fly Routine (Executive CPC)	198
B-31	Data Input for Real Time Control Sequence Routine (Data Input CPC)	202
B-32	Data Input for Programming Panel Thumbwheel Switches Routine (Data Input CPC)	205
B-33	Data Output for Real Time Control Sequence Routine (Data Output CPC)	208

LIST OF ILLUSTRATIONS (Concluded)

Figure		Page
B-34	Reset Watchdog Timer Routine (Data Output CPC)	210
B-35	Format Signal Display Output Command Words Routine (Data Output CPC)	213
B-36	Turn Programming Panel Status Lights On and Off Routines (Data Output CPC)	217
B-37	Timing Plan Execution Monitor Routine (Timing Plan Execution CPC)	222
B-38	Synchronized Start Routine (Timing Plan Execution CPC)	225
B-39	Initiate Next Interval Routine (Timing Plan Execution CPC)	229
B-40	Initiate Next Cycle Routine (Timing Plan Execution CPC)	233
B-41	Copy Timing Plan Tables Routine (Timing Plan Execution CPC)	236
B-42	Bus Preempt Monitor Routine (Bus Preempt CPC)	243
B-43	Bus Preempt Request Routine (Bus Preempt CPC)	250
B-44	Bus Preempt Termination Routine (Bus Preempt CPC)	257
B-45	Vehicle Classification Routine (Bus Preempt CPC)	267
B-46	Preempt Interval Modifier (Bus Preempt CPC)	275
C-1	Preempt Monitor - PDB14C	303

LIST OF TABLES

Table		Page
B-1	Memory Size of Assembled Routines	280
B-2	RAM Data Store/Fetch Matrix	281
B-3	Bit Assignments in the Discrete Input Data Word	286
B-4	Bit Assignments in the Main Street Command Register Status Word	287
B-5	Bit Assignments in the Cross Street Command Register Status Word	288
B-6	Bit Assignments in the Data Input Word for the Programmable Panel Discrete Switches	288
B-7	Sequence of Thumbwheel Switch Data in the Data Base Buffers BPD1, BPD2 and BPD3	289
B-8	Bit Assignments in the Status Display Output Word	289
B-9	Format of the Main Street Output Command Word (0=Off, 1=On)	290
B-10	Format of the Cross Street Output Command Word (0=Off, 1=On)	290
B-11	Format of the Binary-Coded Thumbwheel Data Input Words	291

LIST OF ABBREVIATIONS

BDPP	Bus Detector Preprocessor
BPCU	Bus Preempt Controller Unit
CCU	Computer Control Unit
CMU	Conflict Monitor Unit
CPU	Central Processing Unit
IFU	Interface Electronics Unit
LDU	Light Driver Unit
LMC	Loop Master Controller
LSB	Least Significant Bit
MCU	Microcomputer Unit
MSB	Most Significant Bit
PCIO	Program Controlled Input/Output
PPDU	Programming Panel Discretes Unit
PP1U	Programming Panel Plan 1 Unit
PP2U	Programming Panel Plan 2 Unit
PP3U	Programming Panel Plan 3 Unit
PROM	Programmable Read Only Memory
RAM	Random Access (Read/Write) Memory
RTC	Real Time Clock
SDB	Signature Data Block
SDMU	Signal Light Driver and Monitor Unit
SDU	Status Display Unit
TCP	Transfer Control Panel

APPENDIX A
FORTRAN IV VERSION OF THE VEHICLE CLASSIFICATION ALGORITHM

3.0 PROGRAM DOCUMENTATION

3.0.1 Identification

This document describes the computer program that models the vehicle classification algorithm in the Passive Bus Detector Software. This program is written in the FORTRAN IV language and is intended for operation on a general-purpose computer for off-line analysis of vehicle signatures as processed through the classification algorithm. This document is written according to the FHWA software documentation standard.

3.0.2 Functional Summary

This program consists of the main program and subroutine VCLAS and ERE. The main program reads the input data file, adds the end of signature status to the individual signatures, calls the classifier, VCLAS, for processing of each signature, and prints the summary data at the end of the job.

The classification algorithm in subroutine VCLAS performs all the functions of the algorithm in the passive bus detector program. Each signature is first tested for noise and data validity. Signatures that pass these tests and then classified as a standard transit bus or as not a standard transit bus. Error data are recorded by calling the subroutine, ERE. Classification results are printed for all signatures in the data base with the level of output controlled by the printout control parameter from the input file.

The error recording routine, ERE, records the categories of error data as determined by the classification algorithm. The level of error data printout is also controlled by the printout control parameter.

The symbology and program operation are identical as much as possible with the corresponding functions in the Passive Bus Detector Software.

3.1 MAIN PROGRAM

3.1.1 Program Narrative

3.1.1.1 Purpose

This is the main program in the FORTRAN IV version of the vehicle classification algorithm. This routine initializes data arrays, reads the input data file, appends the end of signature status to each signature, calls the classifier, and prints the summary data for each job.

3.1.1.2 Software Interfaces

The input data file read by this program includes two types of data: the first record contains the printout control parameter, VCOUT, and subsequent records up to the end of the file contain vehicle signature data.

Print Control Parameter, VCOUT

This datum controls the level of output generated by subroutine VCLAS and ERE.

VCOUT = 0 – Summary tables only.

= 1 – Option 0 plus additional information indicating program logic flow through the classifier.

= 2 – Option 1 plus data values computed for decision points in the classifier.

Vehicle Signature Data, SDB(8)

The vehicle signature data are the features that are extracted and digitized from the detector signal by the preprocessor component of the passive bus detector system. These integer data for each signature are contained in one record of the input file read by the main program. All signature records up to the end of file are read and processed in turn.

SBD(1) Total signature duration.

SDB(2) First peak amplitude.

SDB(3) Time from start of signature to first peak.

SDB(4) Second peak amplitude.

SDB(5) Time from start of signature to second peak.

SDB(6) Third or last peak amplitude if the signature has three or more peaks.

SDB(7) Time from start of signature to the third or last peak if the signature has three or more peaks

SDB(8) End of signature status. This data is set to 1 by the main program prior to calling the classifier routine.

Data Scaling

Time values, SDB(1), SDB(3), SDB(5), and SDB(7), have units of 4 milliseconds. For a signature of 800 milliseconds total duration, SDB(1) = 200.

Peak amplitude values are scaled in volts corresponding to a range in phase shift of the signal. However, the classifier uses these values only in ratios, and therefore, the actual scaling is not required in the classifier. The data ranges are given below:

Time values: 12 bits, 0 to 4095

Amplitude values: 8 bits, 0 to 255

Subroutines Called

The main program calls the vehicle classifier, subroutine VCLAS. The signature data are transferred via blank common.

3.1.1.3 Assumptions

All signatures are processed using the signature data base for detector Number 1.

3.1.1.4 Solution Approach

The signature data is read by the main program from the input file. The data is stored in the signature data block for detector Number 1 and the classifier is called to classify the signature as bus or non-bus.

3.1.1.5 Program Operation

The main program first inputs the printout control parameter, VCOUT. Data arrays for error data and classifier results are

cleared to zero. Each subsequent record read from the input file is a vehicle signature, containing seven integer values. The data are stored in the first signature data block and the classifier is called. Following the classification of all the signatures in the input file, the summary data are printed and the job is terminated. The end of file is indicated by a zero value in the first data field of the vehicle signature record.

3.1.1.6 Model

Not applicable.

3.1.1.7 Date

June, 1975

3.1.1.8 Remarks

None

3.1.2 Program Work Areas

Signature Data Base, SDB(32,8)

SDB(VCDT, 1)	Total duration of signature (*4 msec)
SDB(VCDT, 2)	First peak amplitude
SDB(VCDT, 3)	Time from start of signature to first peak (*4 msec)
SDB(VCDT, 4)	Second peak amplitude
SDB(VCDT, 5)	Time from start of signature to second peak (*4 msec)
SDB(VCDT, 6)	Last peak amplitude
SDB(VCDT, 8)	End of signature word - A non-zero value in this word indicates that the signature data is complete and ready for classification.

VCDB(25) Vehicle Classifier Data Base

VCDB(1-7)	Vehicle signature copied from SDB (specified detector, 1-7).

Using, A0 = VCDB(2) and T(C1) = VCDB(3)

A1 = VCDB(4) T(C2) = VCDB(5)

A2 = VCDB(6) T(C3) = VCDB(7)

T(C4) = VCDB(1)

The following is computed for the moving/stopped test:
VCDB(8) T(C1) + T(C4) - T(C3)

The following are computed in the stopped vehicle classifier:

VCDB(9) VC31 * A1

VCDB(10) VC32 * A2

VCDB(11) VC41 * A1

VCDB(12) VC42 * A2

VCDB(13) ((VC32*A1) + (VC32+A2))/A0 (H1 function)

VCDB(14) ((VC41*A1) + (VC42*A2))/A0 (H2 function)

The following are determined in the moving vehicle classifier:

VCDB(15) Ci4*A2 Note: i=1 for K4 function
i=2 for K5 function

VCDB(16) Ci3*A1

VCDB(17) ((Ci3*A1) - (Ci4*A2))/A0

VCDB(18) Ci1*(T(C2) - T(C1))

VCDB(19) Ci2*(T(C3) - T(C2))

VCDB(20) (Ci2*(T(C2) - T(C1)) - Ci1*(T)C3)-T(C2))/
(T(C1)+T(C4) - T(C3))

VCDB(21) K4 function VCDB(17) + VCDB(20) + Ci5

VCDB(22) K5 function VCDB(17) + VCDB(20) + Ci5

VCDB(23-25) Not Used

Vehicle Classifier Vehicle Log, VCLOG(15)

VCLOG(1) Number of signatures rejected as noise (total duration of signature less than 128 milliseconds).

VCLOG(2) Number of signatures that failed gross features test and are thus classified as non-buses (fewer than three peaks).

VCLOG(3) Number of signatures classified by stopped vehicle classifier.

VCLOG(4) Number of signatures classified by moving vehicle classifier.

VCLOG(5) Number of signatures that passed gross features test and were classified as non-buses.

VCLOG(6) Number of signatures classified as buses.

VCLOG(7-15) Not Used

Preempt Error Buffer, PERR(17)

PERR(1) Error flag (not used)

PERR(2) Number of overflow errors detected

PERR(3) Detector index for last overflow error

PERR(4) Number of times specified detector was out of range

PERR(5) Detector index for last such error

PERR(6) Number of times an amplitude value was out of range

PERR(7) Detector index for last such error

PERR(8) Number of times a time value was out of range

PERR(9) Detector index for last such error

PERR(10) Number of times a time value was out of sequence

PERR(11) Detector index for last such error

PERR(12) Number of vehicle signatures rejected as noise

PERR(13) Detector index for last such error

PERR(14) Number of times EOS status word changed during copy operation

PERR(15) Detector index for last such error

PERR(16) Number of times a signature was lost during a copy operation

PERR(17) Detector index for last such error

H1T1 Evaluation term of H1 function (minimum value or bus)

H1T2 Evaluation term for H1 function (maximum value for bus)

H2T1 Evaluation term for H2 function (minimum value for bus)

H2T2 Evaluation term for H2 function (maximum value for bus)

VCHD Highest acceptable detector index

VCLD Lowest acceptable detector index -1

VCOUT Printout control option parameter. Controls the type of output generated by program:

VCOUT = 0 printout of summary information only
= 1 limited printout
= 2 full printout

VEH Set by VCLAS just prior to return to the main program:

VEH = 1 vehicle signature belongs to non-bus
VEH = 0 vehicle signature beongs to bus

VCVC(32) Vehicle Counter. Incremented after a vehicle passes the gross feature test. VCVC(1) is the number of vehicles that have passed over detector i.

Floating Point Constants, CON(33) (Holds the floating point values of the classifier constants. These constants are multiplied by appropriate powers of 2 to get the scaled values required for the classifier computations.)

CCN(1)	VCST1
CON(2)	VCST2
CON(3)	VCST3
CON(4-8)	Used for output operation*
CON(9)	VC31
CON(10)	VC32
CON(11)	VC41
CON(12)	VC42
CON(13-19)	Used for output operation*
CON(20-29)	VCCC (1-10)
CON(30)	H1T1
CON(31)	H1T2
CON(32)	H2T1
CON(33)	H2T2

*The same format statement is used to output the results of the two K functions computation. The coefficients stored in these locations depends on which function, K4 or K5, is being evaluated. These coefficients are printed out with the detailed level of output for print option VCOUT=2.

3.1.3 Program Flowchart

The flowchart of the main program is presented in Figure A-1.

3.1.4 Program Listing

Refer to Appendix D.

3.1.5 Input/Output Record Formats

Input - Two types of input formats are used in the FORTRAN version of the vehicle classifier: the first format uses an I2 format starting on Column 1 to read in the printout option control parameter (VCOUT); the second format uses a 7I4 format to read the vehicle signature data items.

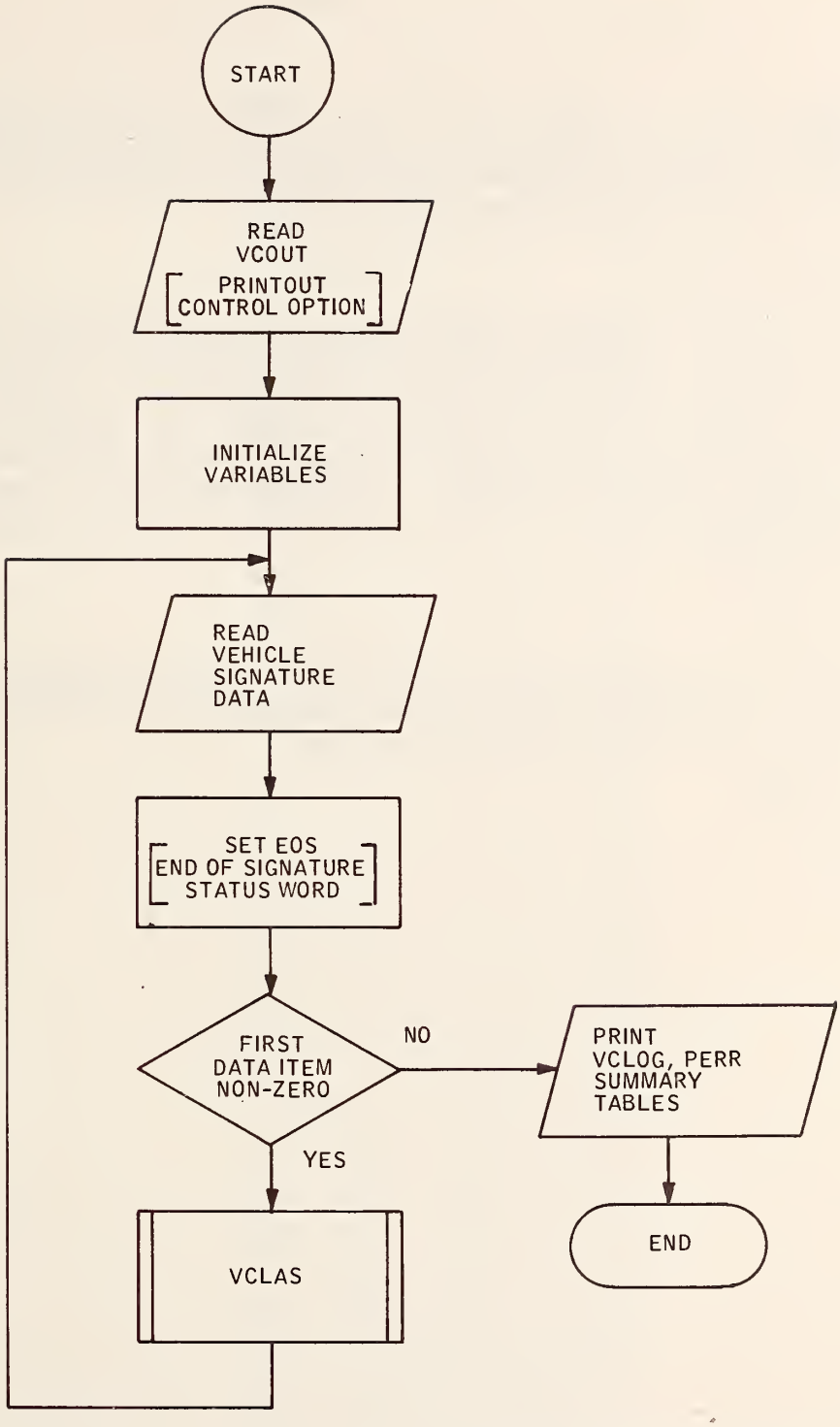


Figure A-1. Flowchart of the FORTRAN IV Vehicle Classifier Main Program

Output - The variables found in the CON array have floating point values and a F8.5 format is employed when outputting these variables. All other variables have integer or scaled integer values that are outputted through an I10 format.

3.1.6 Report Format

One of three available report formats is specified by VCOUT (see 3.1.5). If VCOUT equals zero, the output is limited to summary tables of the error record buffer (PERR) and of the vehicle log (VCLOG). If VCOUT equals one, the output listing is expanded to show whether the moving or stopped vehicle classifier was classified as bus or non-bus. If VCOUT equals two, the output is expanded to include the data values used at decision points through the program.

3.1.7 Test Data

The three data cases given in the following pages illustrate the three levels of optional printout control parameter (VCOUT = 0, 1, 2) in the first record of the data input file.

3.1.8 Revisions

None

APPENDIX A. 3.1.7 TEST DATA

CASE 1

CASE 1 VCBUT = [illegible]

```
00
1000 112 180 146 480 161 548
 512 114 164 147 440 162 780
1344 113 312 152 760 1671196
 780  50 144  67 364  74 676
 628  93 116 120 308 132 536
 688  51 132  65 404  74 600
 432 207 168 155 240 199 296
 392 205 152 120 220 200 260
1484  70 264  44 532  571288
2020  77 356  44 768  66174
1728  75 280  42 644  581496
4095  82 620  321432  274 95
 540 153 168 141 353 153 404
 512 179 136 179 180 172 384
 464 166 148 151 300 164 344
 536 102  96 154 284 133 448
 696  96 124 148 364 130 576
 684 102 120 153 356 133 564
 660  93 116 137 364 123 552
 632  42 156  74 332  64 528
3540  41 220  692804  613396
2576 111 164 1522136 1362448
 692 103 120 158 368 135 576
 716  47 160  77 380  67 604
1920 115 120 1571424 1391788
 768  70 148 171 484  85 660
 772  69 152 168 496  83 668
 880  70 176 171 360  86 756
 960  17 196  47 644  20 876
 912  18 184  47 616  21 828
 464 103 184  93 236  68 364
2308 179 340 102 920 1052096
1872 140 364 121 508  861724
1704 158 316 136 408  98 708
4095 1394095 1834095  754095
4095  581268  674095  604095
 452 109  84 153 220 169 384
4095  861448 1154095  604095
4095  561084  874095  684095
4095 1564095 1464095 1444095
4095  592952  864095  734095
4095  40 980  613532  544095
4095 1114095 1004095  754095
4095 1004095  844095 1204095
4095 1082684  294095  834095
4095  482268  664095  594095
 100  14  3500000000000000000
 100  14  3500000000000000000
 190  15 1260000000000000000
 195  15 1260000000000000000
 197  15 1260000000000000000
8000 109  84 153 220 169 999
 4524000  84 153 220 169 384
 4524000  84 153 220 169 384
 300 109  84 153 220 169 384
 300 109  84 153 220 169 384
 300 109  84 153 220 169 384
 300 109  84 153 220 169 384
0000
```

```
RUN (LMN,PBOVC)
EXECUTION  13:27
```

```
END OF FILE READ
        VEHICLE CLASSIFICATION - VEHICLE LOG
             NOISE                          0
             GROSS TEST FAILURES            5
             STOPPED VEHICLES              19
             MOVING VEHICLES               27
             NON-BUSES                     20
             BUSES                         26



        PREEMPT ERROR BUFFER DUMP
             ERROR FLAG          0
                                                COUNTER     KEY WORD
             OVERFLOW                                1           1
             DETECTOR OUT OF RANGE                   0           0
             AMPLITUDE VALUE OUT OF RANGE            2           1
             TIME VALUE OUT OF RANGE                 1           1
             TIME VALUE OUT OF ORDER                 4           1
             NOISE                                   0           0
             EOS STATUS CHANGE DURING COPY           0           0
             SIGNATURE LOSS                          0           0
*STOP* 0
```

APPENDIX A. 3.1.7 TEST DATA

CASE 2

CASE 2 VCOUT = 1

```
01
1000  112  180  146  480  161  848
 912  114  164  147  440  162  780
1344  113  312  152  760 1671196
 780   50  144   67  364   74  676
 628   93  116  120  308  132  536
 688   51  132   65  404   74  600
 432  207  168  155  240  199  296
 392  205  152  190  220  200  260
1484   70  264   44  532  571288
2020   77  356   44  768   66174-
1728   75  280   42  644  581496
4095   82  620  321432  274095
 540  153  168  141  353  153  404
 512  179  136  179  180  172  384
 464  166  148  151  300  164  344
 536  102   96  154  284  133  448
 696   96  124  148  364  130  576
 684  102  120  153  356  133  564
 660   93  116  137  364  123  552
 632   42  156   74  332   64  528
3540   41  220  692804  613396
2576  111  164 1522136 1362448
 692  103  120  158  368  135  576
 716   47  160   77  380   67  604
1920  115  120 1571424 1391788
 768   70  148  171  484   85  660
 772   69  152  168  496   83  668
 880   70  176  171  560   86  756
 960   17  196   47  644   20  876
 912   18  184   47  616   21  828
 464  103  184   93  236   68  364
2308  179  340  102  920 1052096
1872  140  364  121  508  861724
1704  158  316  136  408   98  708
4095 1394095 1834095  754095
4095  581268  674095  604095
 452  109   84  153  220  169  384
4095  861448 1154095  604095
4095  561084  874095  684095
4095 1564095 1464095 1444095
4095  592952  864095  734095
4095   40 980  613532  544095
4095 1114095 1004095  754095
4095 1004095  844095 1204095
4095 1082684  294095  834095
4095  482268  664095  594095
 100   14   350000000000000000
 100   14   350000000000000000
 190   15 1260000000000000000
 195   15 1260000000000000000
 197   15 1260000000000000000
8000  109   84  153  220  169  999
4524000   84  153  220  169  384
4524000   84  153  220  169  384
 300  109   84  153  220  169  384
 300  109   84  153  220  169  384
 300  109   84  153  220  169  384
 300  109   84  153  220  169  384
0000
```

```
13:28 DEC 31,'99 ID=0006-C00
JOB 4710, PBDVC
ASSIGN M:SI,(DEVICE,CRA03),(FBCD)
ASSIGN F:5,(DEVICE,CRA03),(FBCD)
ASSIGN F:9,(DEVICE,LPA02)
RUN (LMN,PBDVC)
EXECUTION  13:28

     VEHICLE-CLASSIFIER
         MOVING/STOPPED TEST
         MOVING VEHICLE          F0=          8  B0       F1*F2=          32  B0

     BUS


     VEHICLE-CLASSIFIER
         MOVING/STOPPED TEST
         MOVING VEHICLE          F0=          8  B0       F1*F2=          33  B0

     BUS


     VEHICLE-CLASSIFIER
         MOVING/STOPPED TEST
         MOVING VEHICLE          F0=         10  B0       F1*F2=          30  B0

     BUS


     VEHICLE-CLASSIFIER
         MOVING/STOPPED TEST
         MOVING VEHICLE          F0=          9  B0       F1*F2=          34  B0

     BUS


     VEHICLE-CLASSIFIER
         MOVING/STOPPED TEST
         MOVING VEHICLE          F0=          8  B0       F1*F2=          32  B0

     BUS


     VEHICLE-CLASSIFIER
         MOVING/STOPPED TEST
         MOVING VEHICLE          F0=          9  B0       F1*F2=          34  B0
```

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE F0= 8 80 F1*F2= 6 80

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE F0= 4 80 F1*F2= 6 80

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE F0= 9 80 F1*F2= 35 80

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE F0= 8 80 F1*F2= 34 80

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE F0= 80 F1*F2= 38 80

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 80 F1*F2= 80

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 8 80 F1*F2= 12 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 8 80 F1*F2= 15 80

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 8 80 F1*F2= 11 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 5 80 F1*F2= 30 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 8 80 F1*F2= 29 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 8 80 F1*F2= 29 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 8 80 F1*F2= 31 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 9 80 F1*F2= 22 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F[illegible]= 9 8[illegible] F1*F2= 139 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE FO= 5 80 F1*F2= 125 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 8 80 F1*F2= 30 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE FO= 9 80 F1*F2= 26 80

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE FO= 7 80 F1*F2= 105 80

BUS

```
VEHICLE-CLASSIFIER
    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=        9  B0      F1*F2=        32  B0

NON-BUS


VEHICLE-CLASSIFIER
    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=        9  B0      F1*F2=        32  B0

NON-BUS


VEHICLE-CLASSIFIER
    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=        9  B0      F1*F2=        30  B0

NON-BUS


VEHICLE-CLASSIFIER
    MOVING/STOPPED TEST
    STOPPED VEHICLE         F0=       11  B0      F1*F2=         0  B0

    ERROR DETECTED                    1
NON-BUS


VEHICLE-CLASSIFIER
    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=       10  B0      F1*F2=        38  B0

NON-BUS


VEHICLE-CLASSIFIER
    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=       10  B0      F1*F2=        10  B0

NON-BUS


VEHICLE-CLASSIFIER
```

MOVING/STOPPED TEST
STOPPED VEHICLE F0= 9 B0 F1*F2= 5 B0

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 11 B0 F1*F2= 6 B0

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 3 B0 F1*F2= 0 B0

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 14 B0 F1*F2= 0 B0

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 14 B0 F1*F2= 1 B0

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
MOVING VEHICLE F0= 5 B0 F1*F2= 31 B0

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F?- 16 B F1-F2- BO

NON-BUS

VEHICLE-CLASSIFIER
MOVING/STOPPED TEST
STOPPED VEHICLE F0- 1A BC F1-F2- BO

BUS

VEHICLE-CLASSIFIER
GROSS TEST FAILURE

VEHICLE-CLASSIFIER
GROSS TEST FAILURE

VEHICLE-CLASSIFIER
GROSS TEST FAILURE

VEHICLE-CLASSIFIER
GROSS TEST FAILURE

VEHICLE-CLASSIFIER
GROSS TEST FAILURE

VEHICLE-CLASSIFIER
ERROR DETECTED 3 1

VEHICLE-CLASSIFIER
ERROR DETECTED 2 1

VEHICLE-CLASSIFIER
ERROR DETECTED 2 1

VEHICLE-CLASSIFIER

MOVING/STOPPED TEST		
ERROR DETECTED	4	1
VEHICLE-CLASSIFIER		
MOVING/STOPPED TEST		
ERROR DETECTED	4	1
VEHICLE-CLASSIFIER		
MOVING/STOPPED TEST		
ERROR DETECTED	4	1
VEHICLE-CLASSIFIER		
MOVING/STOPPED TEST		
ERROR DETECTED	4	1

```
END OF FILE READ
        VEHICLE CLASSIFICATION - VEHICLE LOG
             NOISE                                  0
             GROSS TEST FAILURES                    5
             STOPPED VEHICLES                      19
             MOVING VEHICLES                       27
             NON-BUSES                             20
             BUSES                                 26



        PREEMPT ERROR BUFFER DUMP
             ERROR FLAG          0
                                                  COUNTER     KEY WORD
             OVERFLOW                                 1           1
             DETECTOR OUT OF RANGE                    0           0
             AMPLITUDE VALUE OUT OF RANGE             2           1
             TIME VALUE OUT OF RANGE                  1           1
             TIME VALUE OUT OF ORDER                  4           1
             NOISE                                    0           0
             EOS STATUS CHANGE DURING COPY            0           0
             SIGNATURE LOSS                           0           0
*STOP* 0
```

CASE 2 VCOUT = 2

```
02
1000 112 180 146 480 161 848
 912 114 164 147 440 162 780
1344 113 312 152 760 1671196
 780  50 144  67 364  74 676
 628  93 116 120 308 132 536
 688  51 132  65 404  74 600
 432 207 168 155 240 199 29[illegible]
 392 205 152 19[illegible] 220 200 26[illegible]
1484  70 264  44 532  571288
2020  77 356  44 768  661740
1728  75 280  42 644  581496
4095  82 620  321432  274095
 540 153 168 141 353 153 404
 512 179 136 179 180 172 384
 464 166 148 151 300 164 344
 536 102  96 154 284 133 448
 696  96 124 148 364 13[illegible] 576
 684 102 120 153 356 133 564
 660  93 116 137 364 123 552
 632  42 156  74 332  64 528
3540  41 220  692804  613396
2576 111 164 1522136 1362448
 692 103 120 158 368 135 576
 716  47 160  77 380  67 604
1920 115 120 1571424 1391788
 768  70 148 171 484  85 66[illegible]
 772  69 152 168 496  83 668
 880  70 176 171 560  86 756
 960  17 196  47 644  20 876
 912  18 184  47 616  21 828
 464 103 184  93 236  68 364
2308 179 340 102 920 1052096
1872 140 364 121 508  861724
1704 158 316 136 408  98 708
4095 1394095 1834095  754095
4095  581268  674095  604095
 452 109  84 153 220 169 384
4095  861448 1154095  604095
4095  561084  874095  684095
4095 1564095 1464095 1444095
4095  592952  864095  734095
4095  40 980  613532  544095
4095 1114095 1004095  754095
4095 1004095  844095 1204095
4095 1082684  294095  834095
4095  482268  664095  594095
 100  14  3500000000000000000
 100  14  3500000000000000000
 190  15 1260000000000000000
 195  15 1260000000000000000
 197  15 1260000000000000000
8000 109  84 153 220 169 999
 4524000  84 153 220 169 384
 4524000  84 153 220 169 384
 300 109  84 153 220 169 384
 300 109  84 153 220 169 384
 300 109  84 153 220 169 384
 300 109  84 153 220 169 384
0000
```

```
13:29 DEC 31,'89 ID=0007-C00
JOB 4710, PBDVC
ASSIGN M:S1,(DEVICE,CRA03),(FBCD)
ASSIGN F:5,(DEVICE,CRA03),(FBCD)
ASSIGN F:9,(DEVICE,LPA02)
RUN (LIN,PBDVC)
EXECUTION  13:29
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    1000  ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     112
        TIME FROM TURN ON TO 1ST PEAK            180  ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    146
        TIME FROM TURN ON TO 2ND PEAK            480  ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      161
        TIME FROM TURN ON TO LAST PEAK           848  ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8  80      F1,F2=          32  80


        T(C1)+T(C4)-T(C3) =          332   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                  624358   B13
          .22800*A1 =                  136218   B13
                 G =                    .4358   B13
          .27700*(T(C2)-T(C1)) =       340200   B13
          .01980*(T(C3)-T(C2)) =        29808   B13
                 F =                     .934   B13
                K4 =                   .1770   B13
                K5 =                       0   B13



        T(C1)+T(C4)-T(C3) =          332   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                 2703673   B13
         2.17000*A1 =                 1297648   B13
                 G =                   .12553   B13
         1.66000*(T(C2)-T(C1)) =      2039700   B13
         1.36000*(T(C3)-T(C2)) =      2049760   B13
                 F =                        30   B13
                K4 =                         0   B13
                K5 =                   .7686   B13


B 13
```

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 912 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 114
TIME FROM TURN ON TO 1ST PEAK 164 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 147
TIME FROM TURN ON TO 2ND PEAK 44[illegible] (* 4 MILLISEC)
LAST PEAK AMPLITUDE 162
TIME FROM TURN ON TO LAST PEAK 73[illegible] (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE F2= 8[illegible] F1*F2= 33[illegible]

T(C1)+T(C4)-T(C3) = 296 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
.94700*A2 = 628236 B13
.22800*A1 = 137151 B13
G = .4307 B13
.27700*(T(C2)-T(C1)) = 312984 B13
.01980*(T(C3)-T(C2)) = 27540 B13
F = .964 B13
K4 = .1749 B13
K5 = 0 B13

T(C1)+T(C4)-T(C3) = 296 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
4.100[illegible]0*A2 = 2720466 B13
2.170[illegible]*A1 = 1306536 B13
G = .12402 B13
1.66000*(T(C2)-T(C1)) = 1876524 B13
1.36000*(T(C3)-T(C2)) = 1893800 B13
F = 58 B13
K4 = 0 B13
K5 = .7507 B13

B S

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    1344 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     113
        TIME FROM TURN ON TO 1ST PEAK            312 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    152
        TIME FROM TURN ON TO 2ND PEAK            760 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      167
        TIME FROM TURN ON TO LAST PEAK          1196 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=        10  80       F1*F2=         30  80


        T(C1)+T(C4)-T(C3) =         460   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                  647626   B13
          .22800*A1 =                  141816   B13
                 G =                    -4476   B13
          .27700*(T(C2)-T(C1)) =       505232   B13
          .01980*(T(C3)-T(C2)) =        35316   B13
                 F =                    -1027   B13
                K4 =                    -1981   B13
                K5 =                        0   B13



        T(C1)+T(C4)-T(C3) =         460   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                 2804431   B13
         2.17000*A1 =                 1350976   B13
                 G =                   -12862   B13
         1.66000*(T(C2)-T(C1)) =      3045952   B13
         1.36000*(T(C3)-T(C2)) =      2428520   B13
                 F =                    -1342   B13
                K4 =                        0   B13
                K5 =                    -9367   B13


BUS
```

VEHICLE-CLASSIFIER

```
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                        750 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                         82
        TIME FROM TURN ON TO 1ST PEAK               144 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                        47
        TIME FROM TURN ON TO 2ND PEAK               364 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                          71
        TIME FROM TURN ON TO LAST PEAK              676 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          *  80      F1*F2=          34  80


        T(C1)+T(C4)-T(C3) =         248   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                  236972   813
          .22800*A1 =                   62511   813
                 G =                   -4489   813
          .87700*(T(C2)-T(C1)) =       249480   813
          .31900*(T(C3)-T(C2)) =        25272   813
                 F =                     904   813
                K4 =                   -1871   813
                K5 =                       0   813



        T(C1)+T(C4)-T(C3) =         248   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                 1242682   813
         2.17000*A1 =                  595496   813
                 G =                  -12943   813
         1.66000*(T(C2)-T(C1)) =      1495780   813
         1.36000*(T(C3)-T(C2)) =      1737840   813
                 F =                     976   813
                K4 =                       0   813
                K5 =                   -7130   813
```

BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    628 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     93
        TIME FROM TURN ON TO 1ST PEAK           116 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   120
        TIME FROM TURN ON TO 2ND PEAK           308 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     132
        TIME FROM TURN ON TO LAST PEAK          536 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8  BO       F1*F2=          32  BO


        T(C1)+T(C4)-T(C3) =          208   BO

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 511896   B13
          .22800*A1 =                 111960   B13
                  G =                  -4300   B13
          .27700*(T(C2)-T(C1)) =      217728   B13
          .01980*(T(C3)-T(C2)) =       18468   B13
                  F =                   -957   B13
                 K4 =                  -1735   B13
                 K5 =                      0   B13


        T(C1)+T(C4)-T(C3) =          208   BO

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                2216676   B13
         2.17000*A1 =                1066560   B13
                  G =                 -12366   B13
         1.66000*(T(C2)-T(C1)) =     1305408   B13
         1.36000*(T(C3)-T(C2)) =     1269960   B13
                  F =                   -170   B13
                 K4 =                      0   B13
                 K5 =                  -7699   B13


BJS
```

VEHICLE-CLASSIFIER

```
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    688 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     51
        TIME FROM TURN ON TO 1ST PEAK           132 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    6[illegible]
        TIME FROM TURN ON TO 2ND PEAK           404 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      74
        TIME FROM TURN ON TO LAST PEAK          600 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          [illegible] 80      [illegible]1*F2=          34  80

        T(C1)+T(C4)-T(C3) =          22[illegible]  80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                  286972   B13
          .22800*A1 =                   60645   B13
                  G =                   -4437   B13
          .27700*(T(C2)-T(C1)) =       308448   B13
          .01920*(T(C3)-T(C2)) =        15876   B13
                  F =                   -1329   B13
                 K4 =                   -2244   B13
                 K5 =                       0   B13


        T(C1)+T(C4)-T(C3) =          22[illegible]  80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                 1242682   B13
         2.17000*A1 =                  577720   B13
                  G =                  -13038   B13
         1.66000*(T(C2)-T(C1)) =      1849328   B13
         1.36000*(T(C3)-T(C2)) =      1091720   B13
                  F =                   -3443   B13
                 K4 =                       0   B13
                 K5 =                  -11644   B13
```

BJS

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 432 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 207
TIME FROM TURN ON TO 1ST PEAK 163 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 155
TIME FROM TURN ON TO 2ND PEAK 240 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 199
TIME FROM TURN ON TO LAST PEAK 296 (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE F0= 8 80 F1*F2= 6 80

T(C4)+T(C4)-T(C3) = 304 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

.94700*A2 =	771722	B13
.22800*A1 =	144615	B13
D =	-3029	B13
.27700*(T(C2)-T(C1)) =	81648	B13
.01980*(T(C3)-T(C2)) =	4536	B13
F =	-253	B13
K4 =	240	B13
K5 =	0	B13

NON-BUS

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 392 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 225
TIME FROM TURN ON TO 1ST PEAK 152 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 190
TIME FROM TURN ON TO 2ND PEAK 221 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 200
TIME FROM TURN ON TO LAST PEAK 240 (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE F0= 3 80 F1*F2= 6 80

T(C1)+T(C4)-T(C3) = 284 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

.94700*A2 =	775600	B13
.22500*A1 =	177270	B13
S =	.2918	B13
.27700*(T(C2)-T(C1)) =	77112	B13
.01950*(T(C3)-T(C2)) =	3240	B13
F =	.260	B13
K4 =	344	B13
K5 =	0	B13

NON-BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    1484  ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      70
        TIME FROM TURN ON TO 1ST PEAK            264  ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     44
        TIME FROM TURN ON TO 2ND PEAK            532  ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       57
        TIME FROM TURN ON TO LAST PEAK          1268  ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          9  80      F1*F2=         35  80


        T(C1)+T(C4)-T(C3) =         460   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .9470*A2 =                221046   B13
          .2280*A1 =                 41052   B13
                 G =                  .2571   B13
          .2770*(T(C2)-T(C1)) =     303912   B13
          .0190*(T(C3)-T(C2)) =      61236   B13
                 F =                  .527   B13
                K4 =                   424   B13
                K5 =                     0   B13



VAN-BUS
```

VEHICLE-CLASSIFIER

```
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    2020 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      77
        TIME FROM TURN ON TO 1ST PEAK            356 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     44
        TIME FROM TURN ON TO 2ND PEAK            768 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       66
        TIME FROM TURN ON TO LAST PEAK          1740 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8   BO        F1*F2=          34   BO


        T(C1)+T(C4)-T(C3) =          636    BO

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700* L =                 255948    B13
          .22800*A1 =                  41052    B13
                  G =                  -2790    B13
          .27700*(T(C2)-T(C1)) =      467208    B13
          .01980*(T(C3)-T(C2)) =       78732    B13
                  F =                   -610    B13
                 K4 =                    122    B13
                 K5 =                      0    B13
```

NON-BUS

```
VEHICLE.CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    1728 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      75
        TIME FROM TURN ON TO 1ST PEAK            280 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     42
        TIME FROM TURN ON TO 2ND PEAK            644 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       59
        TIME FROM TURN ON TO LAST PEAK          1496 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8  80      F1*F2=          38  81


        T(C1)+T(C4)-T(C3) =        512   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                  224924   813
          .22800*A1 =                   39186   813
                  G =                   -2476   813
          .27700*(T(C2)-T(C1)) =        412776   813
          .01980*(T(C3)-T(C2)) =         69012   813
                  F =                    -671   813
                 K4 =                     375   813
                 K5 =                       0   813



NOV.BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      82
        TIME FROM TURN ON TO 1ST PEAK            620 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     32
        TIME FROM TURN ON TO 2ND PEAK           1432 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       27
        TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          FC=        1   BO      F1*F2=          .  BO


        T(C1)+T(C4)-T(C3) =        620   BO

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =     357824   B13
          .87600*A2 =      96876   B13
         2.52000*A1 =          0   B13
          .13900*A2 =          0   B13
                 H1 =       5545   B13
                 H2 =          0   B13
NON-BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    540 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    153
        TIME FROM TURN ON TO 1ST PEAK           168 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   141
        TIME FROM TURN ON TO 2ND PEAK           353 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     153
        TIME FROM TURN ON TO LAST PEAK          404 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          61  80      F1*F2=          12  80


        T(C1)+T(C4)-T(C3) =         304   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 593334   B13
          .22800*A1 =                 131553   B13
                  G =                  -3018   B13
          .27700*(T(C2)-T(C1)) =      209790   B13
          .01980*(T(C3)-T(C2)) =        4131   B13
                  F =                   -676   B13
                 K4 =                   -172   B13
                 K5 =                      0   B13



        T(C1)+T(C4)-T(C3) =         304   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                2569329   B13
         2.17000*A1 =                1253208   B13
                  G =                  -8602   B13
         1.66000*(T(C2)-T(C1)) =     1257815   B13
         1.36000*(T(C3)-T(C2)) =      284070   B13
                  F =                  -3203   B13
                 K4 =                      0   B13
                 K5 =                  -5968   B13


BUS
```

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 512 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 179
TIME FROM TURN ON TO 1ST PEAK 136 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 179
TIME FROM TURN ON TO 2ND PEAK 1[illegible] (* 4 MILLISEC)
LAST PEAK AMPLITUDE 172
TIME FROM TURN ON TO LAST PEAK 384 (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE FC= 80 F1*F2= 15 80

T(C1)+T(C4)-T(C3) = 264 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

.9[illegible]*A2 =	667016	B13
.22800*A1 =	167007	B13
S =	-2793	B13
.27700*(T(C2)-T(C1)) =	49896	B13
.01980*(T(C3)-T(C2)) =	16524	B13
F =	-126	B13
K4 =	603	B13
K5 =	0	B13

NON-BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    464 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    166
        TIME FROM TURN ON TO 1ST PEAK           148 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   151
        TIME FROM TURN ON TO 2ND PEAK           300 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     164
        TIME FROM TURN ON TO LAST PEAK          344 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=        8  30      F1*F2=        11  80


        T(C1)+T(C4)-T(C3) =       268   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .947300*A2 =                 635992   B13
          .228000*A1 =                 140883   B13
                   G =                  .2982   B13
          .277000*(T(C2)-T(C1)) =      172368   B13
          .019800*(T(C3)-T(C2)) =        3564   B13
                   F =                  .629   B13
                  K4 =                   .89   B13
                  K5 =                     0   B13



        T(C1)+T(C4)-T(C3) =       268   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.100000*A2 =                2754052   B13
         2.170000*A1 =                1342088   B13
                   G =                  .8505   B13
         1.660000*(T(C2)-T(C1)) =     1033448   B13
         1.360000*(T(C3)-T(C2)) =      245060   B13
                   F =                  .2941   B13
                  K4 =                     0   B13
                  K5 =                  .6609   B13


BUS
```

```
VEHICLE=CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    536 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    102
        TIME FROM TURN ON TO 1ST PEAK            96 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   154
        TIME FROM TURN ON TO 2ND PEAK           284 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     133
        TIME FROM TURN ON TO LAST PEAK          448 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8  80       F1*F2=          30  80

        T(C1)+T(C4)-T(C3) =        184    80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 515774    B13
          .22800*A1 =                 143682    B13
                  G =                  -3647    B13
          .27700*(T(C2)-T(C1)) =      213192    B13
          .01980*(T(C3)-T(C2)) =       13284    B13
                  F =                  -1086    B13
                 K4 =                  -1211    B13
                 K5 =                      0    B13


        T(C1)+T(C4)-T(C3) =        184    80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                2233469    B13
         2.17000*A1 =                1368752    B13
                  G =                  -8477    B13
         1.66000*(T(C2)-T(C1)) =     1278212    B13
         1.36000*(T(C3)-T(C2)) =      913480    B13
                  F =                  -1982    B13
                 K4 =                      0    B13
                 K5 =                  -5622    B13


BUS
```

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 626 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 96
TIME FROM TURN ON TO 1ST PEAK 124 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 148
TIME FROM TURN ON TO 2ND PEAK 364 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 13
TIME FROM TURN ON TO LAST PEAK 576 (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE FC= 8 80 F1*F2= 29 80

T(C1)+T(C4)-T(C3) = 244 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
.94700*A2 = 504140 813
.22800*A1 = 138084 813
G = -3813 813
.27700*(T(C2)-T(C1)) = 272160 813
.01900*(T(C3)-T(C2)) = 17172 813
F = -1045 813
K4 = -1336 813
K5 = 0 813

T(C1)+T(C4)-T(C3) = 244 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
4.10000*A2 = 2183090 813
2.17000*A1 = 1315424 813
G = -9038 813
1.66000*(T(C2)-T(C1)) = 1631760 813
1.36000*(T(C3)-T(C2)) = 1196640 813
F = -1848 813
K4 = 0 813
K5 = -4049 813

BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    684 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    102
        TIME FROM TURN ON TO 1ST PEAK           120 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           356 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     133
        TIME FROM TURN ON TO LAST PEAK          564 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          6  80       F1*F2=          29  80


        T(C1)+T(C4)-T(C3) =        240   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 515774   B13
          .22800*A1 =                 142749   B13
                  G =                  -3657   B13
          .27700*(T(C2)-T(C1)) =      267624   B13
          .01960*(T(C3)-T(C2)) =       16848   B13
                  F =                  -1044   B13
                 K4 =                  -1179   B13
                 K5 =                      0   B13


        T(C1)+T(C4)-T(C3) =        240   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                2233469   B13
         2.17000*A1 =                1359864   B13
                  G =                  -8564   B13
         1.66000*(T(C2)-T(C1)) =     1604564   B13
         1.36000*(T(C3)-T(C2)) =     1153560   B13
                  F =                  -1858   B13
                 K4 =                      0   B13
                 K5 =                  -5585   B13


BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    660 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     93
        TIME FROM TURN ON TO 1ST PEAK           116 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   137
        TIME FROM TURN ON TO 2ND PEAK           364 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     123
        TIME FROM TURN ON TO LAST PEAK          552 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8  BO      F1*F2=          31  BO

        T(C1)+T(C4)-T(C3) =        224    BO

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 476994   B13
          .22800*A1 =                 127821   B13
                  G =                  -3754   B13
          .27700*(T(C2)-T(C1)) =      281232   B13
          .01980*(T(C3)-T(C2)) =       15223   B13
                  F =                  -1187   B13
                 K4 =                  -1419   B13
                 K5 =                      0   B13


        T(C1)+T(C4)-T(C3) =        224    BO

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                2065539   B13
         2.17000*A1 =                1217656   B13
                  G =                  -9117   B13
         1.66000*(T(C2)-T(C1)) =     1686152   B13
         1.36000*(T(C3)-T(C2)) =     1047160   B13
                  F =                  -2852   B13
                 K4 =                      0   B13
                 K5 =                  -7132   B13


BUS
```

VEHICLE-CLASSIFIER

```
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    632 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     42
        TIME FROM TURN ON TO 1ST PEAK            156 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    74
        TIME FROM TURN ON TO 2ND PEAK            332 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      64
        TIME FROM TURN ON TO LAST PEAK           528 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          FO=          9  80      F1*F2=          22  20


        T(C1)+T(C4)-T(C3) =          260   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                  243192   B13
          .22800*A1 =                   69042   B13
                  G =                    .4265  B13
          .27700*(T(C2)-T(C1)) =       199524   B13
          .01980*(T(C3)-T(C2)) =        15876   B13
                  F =                    .706   B13
                 K4 =                    .1449  B13
                 K5 =                       0   B13



        T(C1)+T(C4)-T(C3) =          260   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                 1074752   B13
         2.17000*A1 =                  657712   B13
                  G =                    .9929  B13
         1.66000*(T(C2)-T(C1)) =      1196624   B13
         1.36000*(T(C3)-T(C2)) =      1091720   B13
                  F =                    .403   B13
                 K4 =                       0   B13
                 K5 =                    .5495  B13
```

BJS

```
VEHICLE-CLASSIFIER

      . VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    3540 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      41
        TIME FROM TURN ON TO 1ST PEAK            220 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     69
        TIME FROM TURN ON TO 2ND PEAK           2804 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       61
        TIME FROM TURN ON TO LAST PEAK          3396 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=          9  80      F1*F2=        139  80


        T(C1)+T(C4)*T(C3) =        364   80

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =     771552   813
          .87600*A2 =     213862   813
         2.52000*A1 =     712149   813
          .13800*A2 =      34709   813
                 H1 =      24156   813
                 H2 =       1621   813
BYE
```

```
VEHICLE=CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    2576 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     111
        TIME FROM TURN ON TO 1ST PEAK            164 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    152
        TIME FROM TURN ON TO 2ND PEAK           2136 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      136
        TIME FROM TURN ON TO LAST PEAK          2448 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=          6  B0      F1*F2=        126  B0

        T(C1)+T(C4)-T(C3) =        292   B0

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =   1699664   B13
          .87600*A2 =    487963   B13
         2.52000*A1 =   1568792   B13
          .13900*A2 =     77384   B13
                 H1 =     19708   B13
                 H2 =     14830   B13
BJS
```

```
VEHICLE•CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    692 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    103
        TIME FROM TURN ON TO 1ST PEAK           120 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   158
        TIME FROM TURN ON TO 2ND PEAK           368 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     135
        TIME FROM TURN ON TO LAST PEAK          576 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8  80       F1*F2=          30  80


        T(C1)+T(C4)-T(C3) =        236   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 523530   B13
          .22800*A1 =                 147414   B13
                  G =                  -3651   B13
          .27700*(T(C2)-T(C1)) =      281232   B13
          .01960*(T(C3)-T(C2)) =       16848   B13
                  F =                  -1120   B13
                 K4 =                  -1249   B13
                 K5 =                      0   B13


        T(C1)+T(C4)-T(C3) =        236   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                2267055   B13
         2.17000*A1 =                1404304   B13
                  G =                  -8376   B13
         1.66000*(T(C2)-T(C1)) =     1686152   B13
         1.36000*(T(C3)-T(C2)) =     1158560   B13
                  F =                  -2235   B13
                 K4 =                      0   B13
                 K5 =                  -5774   B13


BUS
```

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 716 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 47
TIME FROM TURN ON TO 1ST PEAK 160 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 77
TIME FROM TURN ON TO 2ND PEAK 384 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 47
TIME FROM TURN ON TO LAST PEAK 564 (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE F0= 9 80 F1*F2= 25 80

T(C1)+T(C4)-T(C3) = 272 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

.94700*A2 =	259826	B13
.22500*A1 =	71841	B13
G =	-3999	B13
.27700*(T(C2)-T(C1)) =	249480	B13
.01980*(T(C3)-T(C2)) =	18144	B13
F =	.850	B13
K4 =	.1327	B13
K5 =	0	B13

T(C1)+T(C4)-T(C3) = 272 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

4.10000*A2 =	1125131	B13
2.17000*A1 =	484376	B13
G =	-.9377	B13
1.66000*(T(C2)-T(C1)) =	1495780	B13
1.36000*(T(C3)-T(C2)) =	1247680	B13
F =	.912	B13
K4 =	0	B13
K5 =	.5432	B13

BUS

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 1920 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 115
TIME FROM TURN ON TO 1ST PEAK 120 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 157
TIME FROM TURN ON TO 2ND PEAK 1424 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 139
TIME FROM TURN ON TO LAST PEAK 1788 (* 4 MILLISEC)

MOVING/STOPPED TEST
STOPPED VEHICLE F0= 7 80 F1*F2= 105 80

T(C1)+T(C4)+T(C3) = 252 80

THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER

2*73000*A1	=	1755574	813
*87600*A2	=	498732	813
2*52000*A1	=	1620397	813
*13900*A2	=	79091	813
H1	=	19602	813
H2	=	14778	813

BUS

```
VEHICLE.CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    768 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     70
        TIME FROM TURN ON TO 1ST PEAK           148 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   171
        TIME FROM TURN ON TO 2ND PEAK           484 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      88
        TIME FROM TURN ON TO LAST PEAK          660 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          0  B0       F1*F2=          32  B0

        T(C1)+T(C4)-T(C3) =        256   B0

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 329630   B13
          .22800*A1 =                 159543   B13
                  G =                  -2429   B13
          .27700*(T(C2)-T(C1)) =      381024   B13
          .01980*(T(C3)-T(C2)) =       14256   B13
                  F =                  -1432   B13
                 K4 =                   -339   B13
                 K5 =                      0   B13


        T(C1)+T(C4)-T(C3) =        256    0

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                1427405   B13
         2.17000*A1 =                1519848   B13
                  G =                   1320   B13
         1.66000*(T(C2)-T(C1)) =     2284464   B13
         1.36000*(T(C3)-T(C2)) =      980320   B13
                  F =                  -5094   B13
                 K4 =                      0   B13
                 K5 =                   1063   B13


NON.BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    772 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     69
        TIME FROM TURN ON TO 1ST PEAK           152 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   168
        TIME FROM TURN ON TO 2ND PEAK           496 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      53
        TIME FROM TURN ON TO LAST PEAK          668 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          9  80      F1*F2=          32  80


        T(C1)+T(C4)-T(C3) =         256   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 321874   B13
          .22800*A1 =                 156744   B13
                 G =                   -2393   B13
          .27700*(T(C2)-T(C1)) =       390096   B13
          .01960*(T(C3)-T(C2)) =        13932   B13
                 F =                   -1469   B13
                K4 =                    -340   B13
                K5 =                       0   B13



        T(C1)+T(C4)-T(C3) =         256   80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                1393819   B13
         2.17000*A1 =                1493184   B13
                 G =                    1440   B13
         1.66000*(T(C2)-T(C1)) =     2338856   B13
         1.36000*(T(C3)-T(C2)) =      958040   B13
                 F =                   -5393   B13
                K4 =                       0   B13
                K5 =                     884   B13


NON-BUS
```

VEHICLE-CLASSIFIER

```
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                          880 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                           70
        TIME FROM TURN ON TO 1ST PEAK                 176 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                         171
        TIME FROM TURN ON TO 2ND PEAK                 560 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                            86
        TIME FROM TURN ON TO LAST PEAK                756 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          9  80        F1*F2=          30  80

        T(C1)+T(C4)-T(C3) =          300    80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                  333508   B13
          .22800*A1 =                  159543   B13
                G =                     -2465   B13
          .27700*(T(C2)-T(C1)) =       435456   B13
          .01980*(T(C3)-T(C2)) =        15876   B13
                F =                     -1398   B13
               K4 =                      -361   B13
               K5 =                         0   B13


        T(C1)+T(C4)-T(C3) =          300    80

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                 1444198   B13
         2.17000*A1 =                 1519848   B13
                G =                      1080   B13
         1.66000*(T(C2)-T(C1)) =      2610816   B13
         1.36000*(T(C3)-T(C2)) =      1091720   B13
                F =                     -5063   B13
               K4 =                         0   B13
               K5 =                       854   B13


NON-BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    960 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     17
        TIME FROM TURN ON TO 1ST PEAK           196 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    47
        TIME FROM TURN ON TO 2ND PEAK           644 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      20
        TIME FROM TURN ON TO LAST PEAK          876 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=         11  B0      F1*F2=          0  B0


     ERROR DETECTED          0           1
        OVERFLOW

    VCDB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    960 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     17
        TIME FROM TURN ON TO 1ST PEAK           196 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    47
        TIME FROM TURN ON TO 2ND PEAK           644 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      20
        TIME FROM TURN ON TO LAST PEAK          876 ( * 4 MILLISEC)
        T(C1)+T(C4)-T(C3) =         280   B0

      THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =     525554   B13
           .87600*A2 =      71760   B13
          2.52000*A1 =          0   B13
           .13900*A2 =          0   B13
                  H1 =      35136   B13
                  H2 =          0   B13

      THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
           .00000*A2 =                      0   B13
           .00000*A1 =                      0   B13
                   G =                      0   B13
           .00000*(T(C2)-T(C1)) =           0   B13
           .00000*(T(C3)-T(C2)) =           0   B13
                   F =                      0   B13
                  K4 =                      0   B13
                  K5 =                      0   B13



        T(C1)+T(C4)-T(C3) =         280   B0

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =     525554   B13
           .87600*A2 =      71760   B13
```

```
2.52000*A1 =          0    B13
 .13900*A2 =          0    B13
        H1 =      35136    B13
        H2 =          0    B13
N9N=BUS
```

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 912 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 18
TIME FROM TURN ON TO 1ST PEAK 184 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 47
TIME FROM TURN ON TO 2ND PEAK 616 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 21
TIME FROM TURN ON TO LAST PEAK 828 (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE F0= 10 80 F1*F2= 38 40

T(C4)+T(C4)-T(C3) = 268 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

.94700*A2 =	81438	B13
.22500*A1 =	43851	B13
G =	-2088	B13
.27700*(T(C2)-T(C1)) =	489868	B13
.01960*(T(C3)-T(C2)) =	17172	B13
F =	-1763	B13
K4 =	-329	B13
K5 =	0	B13

T(C1)+T(C4)-T(C3) = 264 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

4.10000*A2 =	352653	B13
2.17000*A1 =	417736	B13
G =	3615	B13
1.66000*(T(C2)-T(C1)) =	2937168	B13
1.36000*(T(C3)-T(C2)) =	1180840	B13
F =	-6553	B13
K4 =	0	B13
K5 =	1899	B13

VAN-BUS

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 464 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 103
TIME FROM TURN ON TO 1ST PEAK 184 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 93
TIME FROM TURN ON TO 2ND PEAK 236 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 68
TIME FROM TURN ON TO LAST PEAK 364 (* 4 MILLISEC)

MOVING/STOPPED TEST
MOVING VEHICLE F0= 10 80 F1*F2= 10 80

T(C1)+T(C4)-T(C3) = 284 80

THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER

.94700*A2 =	263704	B13
.22800*A1 =	96769	B13
B =	-1717	B13
.27700*(T(C2)-T(C1)) =	58968	B13
.01960*(T(C3)-T(C2)) =	10368	B13
F =	-171	B13
K4 =	1634	B13
K5 =	0	B13

NON-BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    2308 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     179
        TIME FROM TURN ON TO 1ST PEAK            940 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    102
        TIME FROM TURN ON TO 2ND PEAK            920 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      105
        TIME FROM TURN ON TO LAST PEAK          2096 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=          9  80      F1*F2=          50  8

        T(C1)+T(C4)-T(C3) =        552    80

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =    1140564    813
          .87600*A2 =     376740    813
         2.52000*A1 =          0    813
          .13900*A2 =          0    813
                 H1 =       8476    813
                 H2 =          0    813
NON-BUS
```

```
VEHICLE=CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    1872 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     140
        TIME FROM TURN ON TO 1ST PEAK            364 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    121
        TIME FROM TURN ON TO 2ND PEAK            508 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       86
        TIME FROM TURN ON TO LAST PEAK          1724 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=        11  BO       F1*F2=          BO

        T(C1)+T(C4)-T(C3) =        512   BO

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =    1353022   B13
          .87600*A2 =     308568   B13
         2.52000*A1 =          0   B13
          .13900*A2 =          0   B13
                 H1 =      11868   B13
                 H2 =          0   B13
NON=BUS
```

```
VEHICLE-CLASSIFIER

          VEHICLE SIGNATURE
          TOTAL SIGNATURE TIME                    1704 ( * 4 MILLISEC)
          FIRST PEAK AMPLITUDE                     158
          TIME FROM TURN ON TO 1ST PEAK            316 ( * 4 MILLISEC)
          SECOND PEAK AMPLITUDE                    136
          TIME FROM TURN ON TO 2ND PEAK            408 ( * 4 MILLISEC)
          LAST PEAK AMPLITUDE                       93
          TIME FROM TURN ON TO LAST PEAK           708 ( * 4 MILLISEC)

     MOVING/STOPPED TEST
     STOPPED VEHICLE          F0=          3  80      F1*F2=          0  80


          T(C1)*T(C4)-T(C3) =       1312    80

          THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
           2.73000*A1 =    1560752    813
            .87600*A2 =     351624    813
           2.52000*A1 =          0    813
            .139 0*A2 =          0    813
                  H1 =      11850    813
                  H2 =          0    813
NON-BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     139
        TIME FROM TURN ON TO 1ST PEAK           4095 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    183
        TIME FROM TURN ON TO 2ND PEAK           4095 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       75
        TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=        1.  B0      F1*F2=            B0


        T(C1)+T(C4)-T(C3) =       4095    B0

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =    2046306    B13
          .87600*A2 =     269100    B13
         2.52000*A1 =    1888743    B13
          .13900*A2 =      42675    B13
                 H1 =      16657    B13
                 H2 =      13895    B13
BUS
```

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE

TOTAL SIGNATURE TIME	4095	(* 4 MILLISEC)
FIRST PEAK AMPLITUDE	68	
TIME FROM TURN ON TO 1ST PEAK	1268	(* 4 MILLISEC)
SECOND PEAK AMPLITUDE	67	
TIME FROM TURN ON TO 2ND PEAK	4095	(* 4 MILLISEC)
LAST PEAK AMPLITUDE	62	
TIME FROM TURN ON TO LAST PEAK	4095	(* 4 MILLISEC)

MOVING/STOPPED TEST

STOPPED VEHICLE F0= 1. 80 F1*F2= . 80

T(C1)+T(C4)-T(C3) = 1268 80

THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER

2.73000*A1 =	74.9194	B13
.87600*A2 =	2152.0	B13
2.52000*A1 =	6915.7	B13
.13900*A2 =	341.40	B13
H1 =	16625	B13
H2 =	12511	B13

BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    452 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    MOVING VEHICLE          F0=          8  BO       F1*F2=          31  BO


        T(C1)+T(C4)-T(C3) =        152   BO

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .94700*A2 =                 655382   B13
          .22800*A1 =                 142749   B13
                  G =                  -4703   B13
          .27700*(T(C2)-T(C1)) =      154224   B13
          .01900*(T(C3)-T(C2)) =       13284   B13
                 F =                    -927   B13
                 K4 =                  -2108   B13
                 K5 =                      0   B13


        T(C1)+T(C4)-T(C3) =        152   BO

        THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
         4.10000*A2 =                2838017   B13
         2.17000*A1 =                1359864   B13
                  G =                 -13561   B13
         1.66000*(T(C2)-T(C1)) =      924664   B13
         1.36000*(T(C3)-T(C2)) =      913480   B13
                  F =                     -73   B13
                 K4 =                       0   B13
                 K5 =                   -8797   B13


BUS
```

```
VEHICLE-CLASSIFIER

          VEHICLE SIGNATURE
          TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
          FIRST PEAK AMPLITUDE                      86
          TIME FROM TURN ON TO 1ST PEAK           1448 ( * 4 MILLISEC)
          SECOND PEAK AMPLITUDE                    115
          TIME FROM TURN ON TO 2ND PEAK           4095 ( * 4 MILLISEC)
          LAST PEAK AMPLITUDE                       60
          TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

     MOVING/STOPPED TEST
     STOPPED VEHICLE          F0=        16  80      F1*F2=          ) 80


          T(C1)+T(C4)-T(C3) =       1448    80

          THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
           2.730000*A1 =    1235930    813
            .876000*A2 =     215280    813
           2.520000*A1 =    1156915    813
            .139000*A2 =      34140    813
                    H1 =      17455    813
                    H2 =      14198    813
BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      56
        TIME FROM TURN ON TO 1ST PEAK           1084 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     37
        TIME FROM TURN ON TO 2ND PEAK           4095 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       48
        TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=        1[illegible]  BO      F1*F2=          [illegible]  BO


        T(C1)+T(C4)-T(C3) =       1084   BO

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =     972834   B13
          .87600*A2 =     243984   B13
         2.52000*A1 =     897927   B13
          .13900*A2 =      38692   B13
                 H1 =      21728   B13
                 H2 =      16725   B13
B S
```

VEHICLE•CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 4095 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 156
TIME FROM TURN ON TO 1ST PEAK 4095 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 146
TIME FROM TURN ON TO 2ND PEAK 4095 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 144
TIME FROM TURN ON TO LAST PEAK 4095 (* 4 MILLISEC)

MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 80 F1•F2=) 80

T(C1)+T(C4)-T(C3) = 4095 80

THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER

2•73000•A1	=	1632572	813
•87600•A2	=	516672	813
2•52000•A1	=	[illegible]	813
•1391[illegible]•A2	=	0	813
	H1 =	13777	813
	H2 =	[illegible]	813

VAN•BUS

VEHICLE-CLASSIFIER

```
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      59
        TIME FROM TURN ON TO 1ST PEAK           2952 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     86
        TIME FROM TURN ON TO 2ND PEAK           4095 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       73
        TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=        16  80      F1*F2=          0  80


        T(C1)+T(C4)-T(C3) =       2952   80

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.730000*A1 =     961652   813
          .876000*A2 =     261924   813
         2.520000*A1 =     887606   813
          .139000*A2 =      41537   813
                  H1 =      20738   813
                  H2 =      15745   813
```

BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      40
        TIME FROM TURN ON TO 1ST PEAK            980 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     61
        TIME FROM TURN ON TO 2ND PEAK           3532 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       54
        TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=         16  B0      F1*F2=          0  B0


        T(C1)+T(C4)-T(C3) =         98      B0

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =     682102    B13
          .87600*A2 =     193752    B13
         2.52000*A1 =     629581    B13
          .13900*A2 =      30726    B13
                H1 =      21896    B13
                H2 =      16507    B13
BUS
```

VEHICLE=CLASSIFIER

VEHICLE SIGNATURE
TOTAL SIGNATURE TIME 4095 (* 4 MILLISEC)
FIRST PEAK AMPLITUDE 111
TIME FROM TURN ON TO 1ST PEAK 4095 (* 4 MILLISEC)
SECOND PEAK AMPLITUDE 100
TIME FROM TURN ON TO 2ND PEAK 4095 (* 4 MILLISEC)
LAST PEAK AMPLITUDE 7?
TIME FROM TURN ON TO LAST PEAK 4095 (* 4 MILLISEC)

MOVING/STOPPED TEST
STOPPED VEHICLE F0= 16 B0 F1*F2= 0 B0

T(C1)+T(C4)-T(C3) = 4095 B0

THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER

2.73000*A1	=	1118200	B13
.87600*A2	=	2b9100	B13
2.52000*A1	=	0	B13
.13900*A2	=	0	B13
H1	=	12498	B13
H2	=	0	B13

NON-BUS

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    4095  ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     100
        TIME FROM TURN ON TO 1ST PEAK           4095  ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     84
        TIME FROM TURN ON TO 2ND PEAK           4095  ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      120
        TIME FROM TURN ON TO LAST PEAK          4095  ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=        16  B0      F1*F2=          0  B0


        T(C1)+T(C4)-T(C3) =       4095   B0

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =     939288   B13
          .87600*A2 =     430560   B13
         2.52000*A1 =          0   B13
          .13900*A2 =          0   B13
                 H1 =      13698   B13
                 H2 =          0   B13
VAN-BUS
```

```
VEHICLE-CLASSIFIER

          VEHICLE SIGNATURE
          TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
          FIRST PEAK AMPLITUDE                     108
          TIME FROM TURN ON TO 1ST PEAK           2684 ( * 4 MILLISEC)
          SECOND PEAK AMPLITUDE                     29
          TIME FROM TURN ON TO 2ND PEAK           4095 ( * 4 MILLISEC)
          LAST PEAK AMPLITUDE                       83
          TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

     MOVING/STOPPED TEST
     STOPPED VEHICLE          F0=        16  80      F1*F2=          0  80


          T(C1)+T(C4)-T(C3) =       2684   80

          THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
           2.73000*A1 =     324278   B13
            .87600*A2 =     297804   B13
           2.52000*A1 =          0   B13
            .13900*A2 =          0   B13
                   H1 =       5760   B13
                   H2 =          0   B13
NON-BUS
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    4095 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                      48
        TIME FROM TURN ON TO 1ST PEAK           2268 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                     66
        TIME FROM TURN ON TO 2ND PEAK           4095 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                       59
        TIME FROM TURN ON TO LAST PEAK          4095 ( * 4 MILLISEC)

    MOVING/STOPPED TEST
    STOPPED VEHICLE          F0=        16  B0      F1*F2=          0  B0

        T(C1)+T(C4)-T(C3) =      2268   B0

        THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =    738012   B13
          .87600*A2 =    211692   B13
         2.52000*A1 =    681186   B13
          .13900*A2 =     33571   B13
                H1 =     19785   B13
                H2 =     14890   B13
BJS
```

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE		
TOTAL SIGNATURE TIME	100	(* 4 MILLISEC)
FIRST PEAK AMPLITUDE	14	
TIME FROM TURN ON TO 1ST PEAK	35	(* 4 MILLISEC)
SECOND PEAK AMPLITUDE	[illegible]	
TIME FROM TURN ON TO 2ND PEAK	[illegible]	(* 4 MILLISEC)
LAST PEAK AMPLITUDE	[illegible]	
TIME FROM TURN ON TO LAST PEAK	0	(* 4 MILLISEC)

GROSS TEST FAILURE

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE		
TOTAL SIGNATURE TIME	100	(* 4 MILLISEC)
FIRST PEAK AMPLITUDE	14	
TIME FROM TURN ON TO 1ST PEAK	35	(* 4 MILLISEC)
SECOND PEAK AMPLITUDE	0	
TIME FROM TURN ON TO 2ND PEAK	0	(* 4 MILLISEC)
LAST PEAK AMPLITUDE	0	
TIME FROM TURN ON TO LAST PEAK	0	(* 4 MILLISEC)

GROSS TEST FAILURE

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE		
TOTAL SIGNATURE TIME	190	(* 4 MILLISEC)
FIRST PEAK AMPLITUDE	15	
TIME FROM TURN ON TO 1ST PEAK	126	(* 4 MILLISEC)
SECOND PEAK AMPLITUDE	0	
TIME FROM TURN ON TO 2ND PEAK	0	(* 4 MILLISEC)
LAST PEAK AMPLITUDE	0	
TIME FROM TURN ON TO LAST PEAK	0	(* 4 MILLISEC)

GROSS TEST FAILURE

VEHICLE•CLASSIFIER

VEHICLE SIGNATURE		
TOTAL SIGNATURE TIME	195	(* 4 MILLISEC)
FIRST PEAK AMPLITUDE	15	
TIME FROM TURN ON TO 1ST PEAK	126	(* 4 MILLISEC)
SECOND PEAK AMPLITUDE	0	
TIME FROM TURN ON TO 2ND PEAK	0	(* 4 MILLISEC)
LAST PEAK AMPLITUDE	0	
TIME FROM TURN ON TO LAST PEAK	0	(* 4 MILLISEC)

GROSS TEST FAILURE

VEHICLE-CLASSIFIER

VEHICLE SIGNATURE		
TOTAL SIGNATURE TIME	197	(* 4 MILLISEC)
FIRST PEAK AMPLITUDE	15	
TIME FROM TURN ON TO 1ST PEAK	126	(* 4 MILLISEC)
SECOND PEAK AMPLITUDE	0	
TIME FROM TURN ON TO 2ND PEAK	0	(* 4 MILLISEC)
LAST PEAK AMPLITUDE	0	
TIME FROM TURN ON TO LAST PEAK	0	(* 4 MILLISEC)

GROSS TEST FAILURE

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    8000 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     109
        TIME FROM TURN ON TO 1ST PEAK             84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    153
        TIME FROM TURN ON TO 2ND PEAK            220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      169
        TIME FROM TURN ON TO LAST PEAK           999 ( * 4 MILLISEC)


    ERROR DETECTED          3          1
        TIME VALUE OUT OF RANGE

    VCOB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    8000 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                     109
        TIME FROM TURN ON TO 1ST PEAK             84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                    153
        TIME FROM TURN ON TO 2ND PEAK            220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                      169
        TIME FROM TURN ON TO LAST PEAK           999 ( * 4 MILLISEC)
        T(C1)+T(C4)-T(C3) =          1    B0

    THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
         2.73000*A1 =                0    B13
          .87600*A2 =                0    B13
         2.52000*A1 =                0    B13
          .13900*A2 =                0    B13
                 H1 =                0    B13
                 H2 =                0    B13

    THE FOLLOWING ARE DETERMINED IN THE MSVING VEHICLE CLASSIFIER
          .00000*A2 =                     0    B13
          .00000*A1 =                     0    B13
                  G =                     0    B13
          .00000*(T(C2)-T(C1)) =          0    B13
          .00000*(T(C3)-T(C2)) =          0    B13
                  F =                     0    B13
                 K4 =                     0    B13
                 K5 =                     0    B13
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    452 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                   4000
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)


     ERROR DETECTED          2          1
        AMPLITUDE VALUE OUT OF RANGE


     VCOB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    452 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                   4000
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)
        T(C1)+T(C4)-T(C3) =          1    B0

     THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =           0    B13
           .87600*A2 =           0    B13
          2.52000*A1 =           0    B13
           .13900*A2 =           0    B13
                  H1 =           0    B13
                  H2 =           0    B13

     THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
           .00000*A2 =                   0    B13
           .00000*A1 =                   0    B13
                   G =                   0    B13
           .00000*(T(C2)-T(C1)) =        0    B13
           .00000*(T(C3)-T(C2)) =        0    B13
                   F =                   0    B13
                  K4 =                   0    B13
                  K5 =                   0    B13
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    452 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                   4000
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)


    ERROR DETECTED          2          1
        AMPLITUDE VALUE OUT OF RANGE


    VOOB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    452 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                   4000
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)
        T(C1)+T(C4)-T(C3) =          1    B0

      THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =               0    B13
           .87600*A2 =               0    B13
          2.52000*A1 =               0    B13
           .13900*A2 =               0    B13
                  H1 =               0    B13
                  H2 =               0    B13

      THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
           .00000*A2 =                         0    B13
           .00000*A1 =                         0    B13
                   G =                         0    B13
           .00000*(T(C2)-T(C1)) =              0    B13
           .00000*(T(C3)-T(C2)) =              0    B13
                   F =                         0    B13
                  K4 =                         0    B13
                  K5 =                         0    B13
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)

    MOVING/STOPPED TEST

     ERROR DETECTED          4          1
        TIME VALUE OUT OF ORDER

    VCDB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)
        T(C1)+T(C4)-T(C3) =          1    80

      THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =               0    B13
           .87600*A2 =               0    B13
          2.52000*A1 =               0    B13
           .13900*A2 =               0    B13
                  H1 =               0    B13
                  H2 =               0    B13

      THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
           .00000*A2 =                     0    B13
           .00000*A1 =                     0    B13
                   G =                     0    B13
           .00000*(T(C2)-T(C1)) =          0    B13
           .00000*(T(C3)-T(C2)) =          0    B13
                   F =                     0    B13
                  K4 =                     0    B13
                  K5 =                     0    B13
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)

    MOVING/STOPPED TEST

     ERROR DETECTED          4          1
        TIME VALUE OUT OF ORDER

    VCDB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)
        T(C1)+T(C4)-T(C3) =          1    80

      THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =          0    B13
           .87600*A2 =          0    B13
          2.52000*A1 =          0    B13
           .13900*A2 =          0    B13
                  H1 =          0    B13
                  H2 =          0    B13

      THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
           .00000*A2 =                    0    B13
           .00000*A1 =                    0    B13
                   G =                    0    B13
           .00000*(T(C2)-T(C1)) =         0    B13
           .00000*(T(C3)-T(C2)) =         0    B13
                   F =                    0    B13
                  K4 =                    0    B13
                  K5 =                    0    B13
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)

    MOVING/STOPPED TEST

     ERROR DETECTED          4          1
        TIME VALUE OUT OF ORDER

    VOOB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)
        T(C1)+T(C4)=T(C3) =          1    80

      THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =              0    B13
           .27600*A2 =              0    B13
          2.52000*A1 =              0    B13
           .13900*A2 =              0    B13
                  H1 =              0    B13
                  H2 =              0    B13

      THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
          .00000*A2 =                        0    B13
          .00000*A1 =                        0    B13
                  J =                        0    B13
          .00000*(T(C2)-T(C1)) =             0    B13
          .00000*(T(C3)-T(C2)) =             0    B13
                  F =                        0    B13
                 K4 =                        0    B13
                 K5 =                        0    B13
```

```
VEHICLE-CLASSIFIER

        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)

    MOVING/STOPPED TEST

     ERROR DETECTED          4          1
       TIME VALUE OUT OF BORDER

    VCDB ARRAY
        VEHICLE SIGNATURE
        TOTAL SIGNATURE TIME                    300 ( * 4 MILLISEC)
        FIRST PEAK AMPLITUDE                    109
        TIME FROM TURN ON TO 1ST PEAK            84 ( * 4 MILLISEC)
        SECOND PEAK AMPLITUDE                   153
        TIME FROM TURN ON TO 2ND PEAK           220 ( * 4 MILLISEC)
        LAST PEAK AMPLITUDE                     169
        TIME FROM TURN ON TO LAST PEAK          384 ( * 4 MILLISEC)
        T(C1)+T(C4)-T(C3) =          1    80

      THE FOLLOWING ARE DETERMINED IN THE STOPPED VEHICLE CLASSIFIER
          2.73000*A1 =               0    B13
           .87600*A2 =               0    B13
          2.52000*A1 =               0    B13
           .13900*A2 =               0    B13
                  H1 =               0    B13
                  H2 =               0    B13

      THE FOLLOWING ARE DETERMINED IN THE MOVING VEHICLE CLASSIFIER
           .00000*A2 =                         0    B13
           .00000*A1 =                         0    B13
                   G =                         0    B13
           .00000*(T(C2)-T(C1)) =              0    B13
           .00000*(T(C3)-T(C2)) =              0    B13
                   F =                         0    B13
                  K4 =                         0    B13
                  K5 =                         0    B13
```

```
END OF FILE READ
        VEHICLE CLASSIFICATION - VEHICLE LOG
             NOISE                          0
             GROSS TEST FAILURES            5
             STOPPED VEHICLES              19
             MOVING VEHICLES               27
             VAN-BUSES                     20
             BUSES                         26



        PREEMPT ERROR BUFFER DUMP
             ERROR FLAG          0
                                                  COUNTER     KEY WORD
             OVERFLOW                                   1            1
             DETECTOR OUT OF RANGE                      0            0
             AMPLITUDE VALUE OUT OF RANGE               2            1
             TIME VALUE OUT OF RANGE                    1            1
             TIME VALUE OUT OF ORDER                    4            1
             NOISE                                      0            0
             FBS STATUS CHANGE DURING COPY              0            0
             SIGNATURE LOSS                             0            0
*STOP* 0
```

3.2 SUBROUTINE VCLAS

3.2.1 Program Narrative

3.2.1.1 Purpose

This subroutine contains the FORTRAN version of the vehicle classification algorithm of the passive bus detector software and additional printout functions to promote detailed analysis of the classification of vehicle signatures. Vehicle signatures are classified as a standard transit bus or as not a standard transit bus.

3.2.1.2 Software Interfaces

This routine is called by the main program. This routine calls the error recording routine, ERE, as in the passive bus detector software.

The main program communicates the vehicle signature and the print control parameter through blank common.

Error data are communicated to ERE through the call arguments:

I - Index of the error condition

J - Error datum to be stored

K - Location of the error buffer

Additional data for prinout are transferred through blank common.

3.2.1.3 Assumptions

The vehicle classification algorithm distinguishes between the standard 40-foot transit bus and all other vehicles. Signatures of total duration less than 32 time intervals (*4 msec = 128 msec) are assumed to be noise.

3.2.1.4 Solution Approach

This routine performs the same functions as the passive bus detector software algorithm with the addition of printout functions for detailed analysis of the classification of individual vehicle signatures.

3.2.1.5 Program Operation

The classifier data base is cleared and the detector index is checked for being within the proper range. The end of signature status is tested for non-zero, indicating a completed signature. The signature is then copied to the classifier data base and end of signature status is rechecked. This function in the direct memory access environment of the passive bus detector software ensures the integrity of the copied vehicle signature data. The signature is then tested against the noise threshold. The third peak is tested for a non-zero amplitude as the gross test for a bus signature. The amplitude data are then checked for the range of 0 to 255, and time interval data are checked for the range 0 to 4095. The time values are also tested for being in an increasing sequence. If the signature duration has reached the maximum duration, the signature is classified by the stopped vehicle classifier. Other signatures are tested in terms of normalized combinations of the time intervals to determine if the shape of the signature is that of a moving bus signature. If not, the signature is also processed by the stopped vehicle classifier. If the shape is consistent with a uniformly moving signature, the signature is classified by the moving vehicle classifier. Classification results are stored in the vehicle classifier data base. The bus or non-bus result is printed for all signatures under printout options 1 and 2.

For each error condition tested and found by the classifier, the error routine, ERE, is called to record the error data.

3.2.1.6 Model

The vehicle classifier is the equivalent of the classifier in the passive bus detector software.

3.2.1.7 Date

June, 1975

3.2.1.8 Remarks

The program operation and symbology has been kept identical as much as possible to the same function in the passive bus detector software.

3.2.2 Program Work Areas

All data are described in the corresponding section of the main program documentation.

3.2.3 Program Flowchart

The flowchart of subroutine VCLAS is presented in Figure A-2.

3.2.4 Program Listing

Refer to Appendix D.

3.2.5 Input/Output Record Formats

Refer to the main program documentation.

3.2.6 Report Formats

The three output options are presented as test data in that section of the main program documentation.

3.2.7 Test Data

Refer to the test data section for the main program.

3.2.8 Revisions

None.

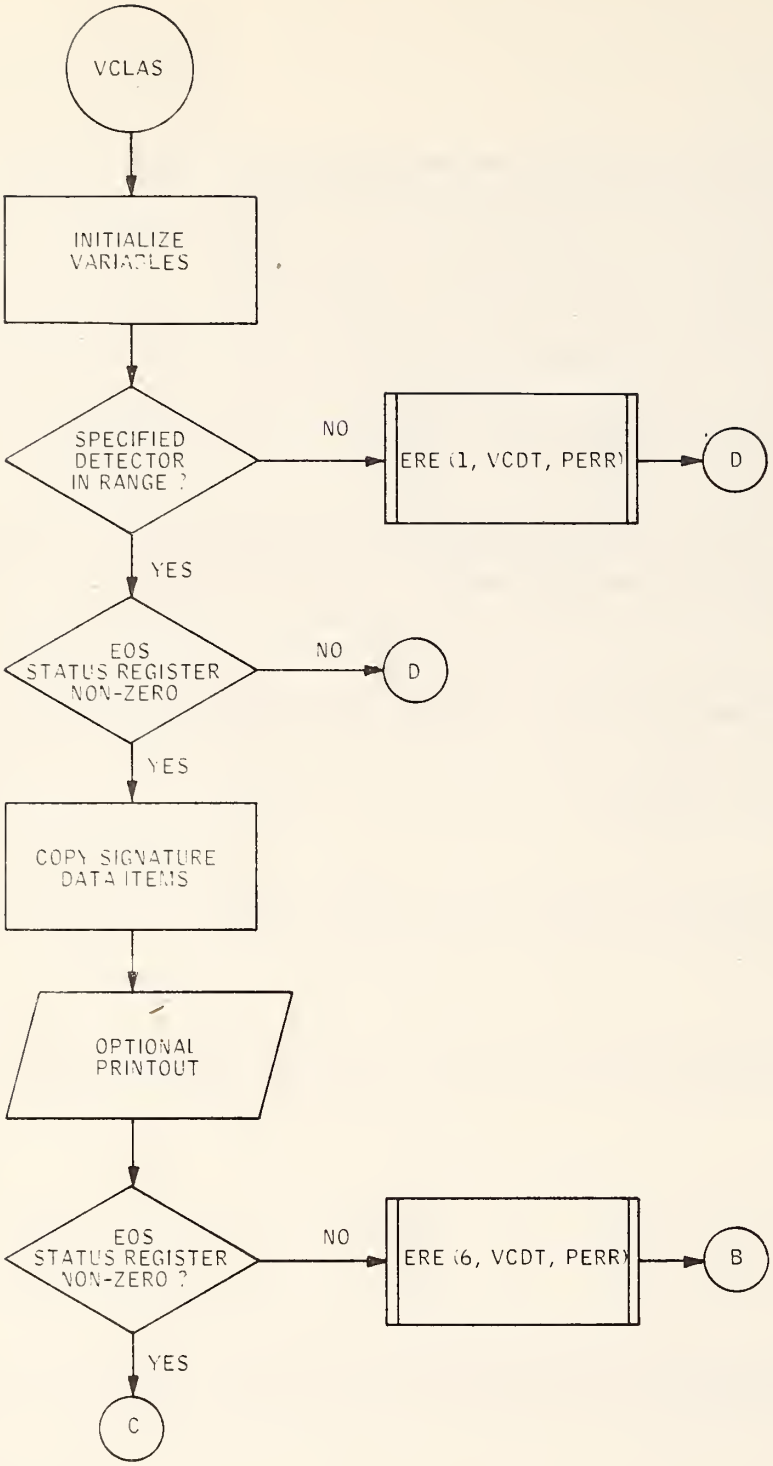


Figure A-2. Flowchart of the FORTRAN IV Vehicle Classifier Vehicle Classification Routine (VCLAS)

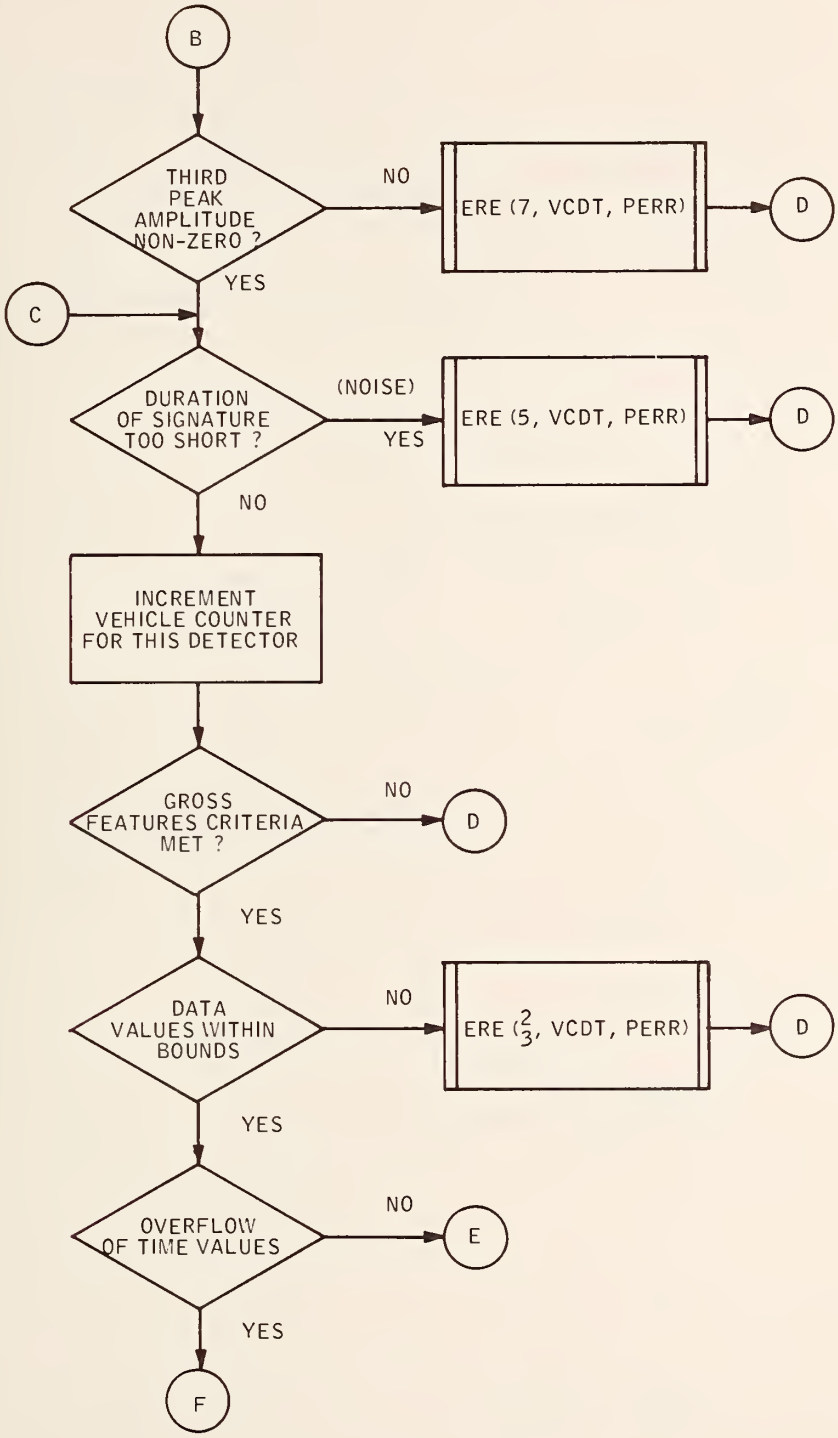


Figure A-2. Flowchart of the FORTRAN IV Vehicle Classifier Vehicle Classification Routine (VCLAS) (Continued)

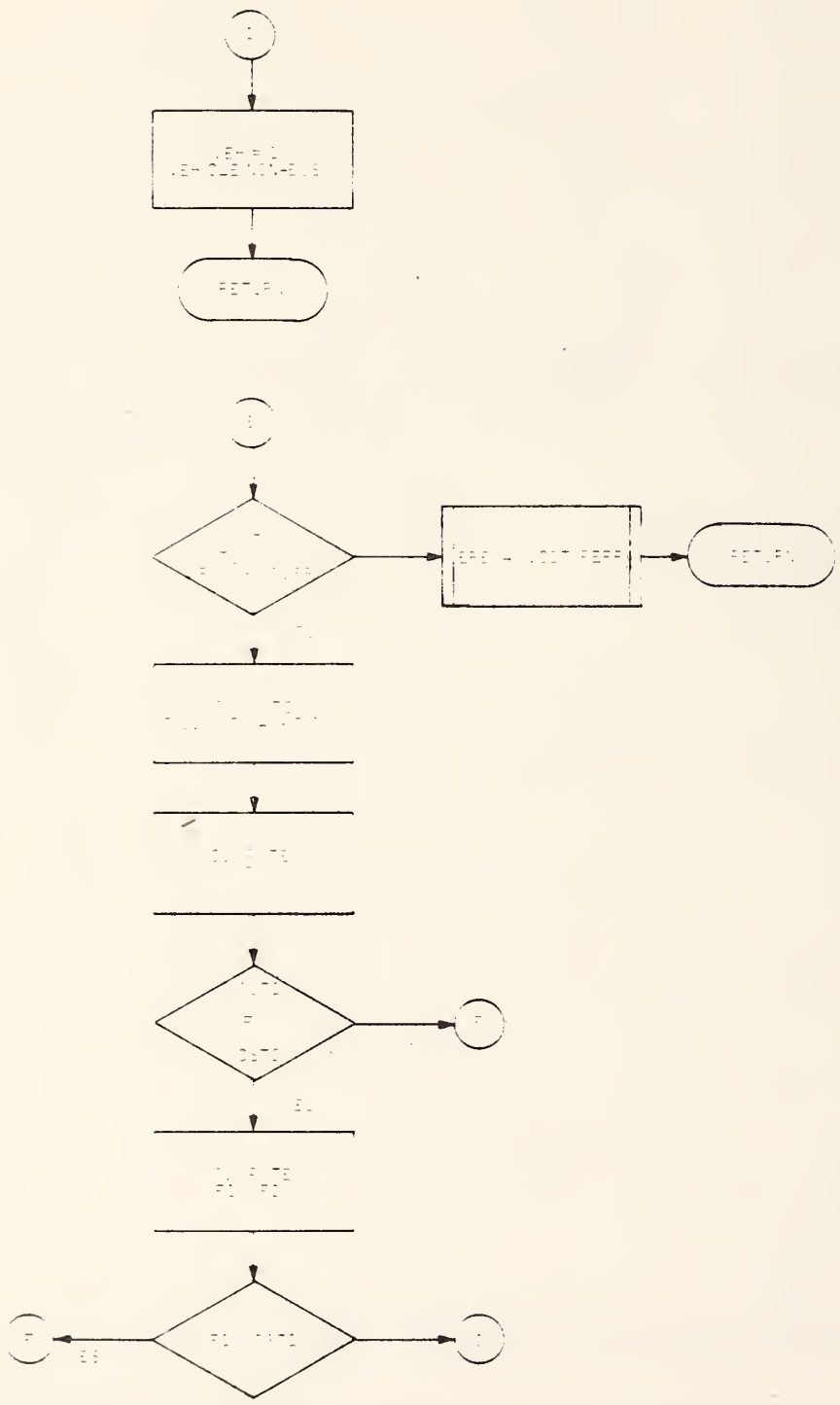


Figure A-2. Flowchart of the FORTRAN IV Vehicle Classifier Vehicle Classification Routine (VCLAS) (Continued)

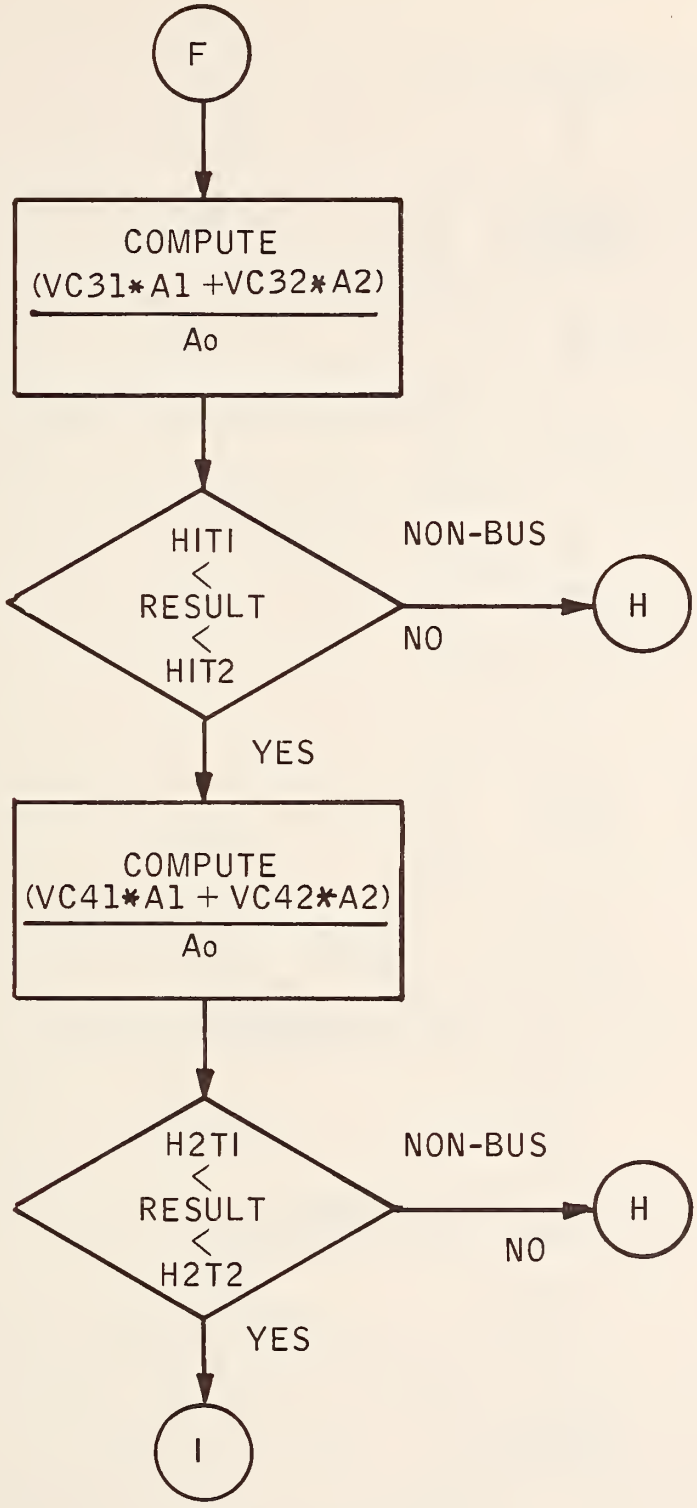


Figure A-2. Flowchart of the FORTRAN IV Vehicle Classifier Vehicle Classification Routine (VCLAS) (Continued)

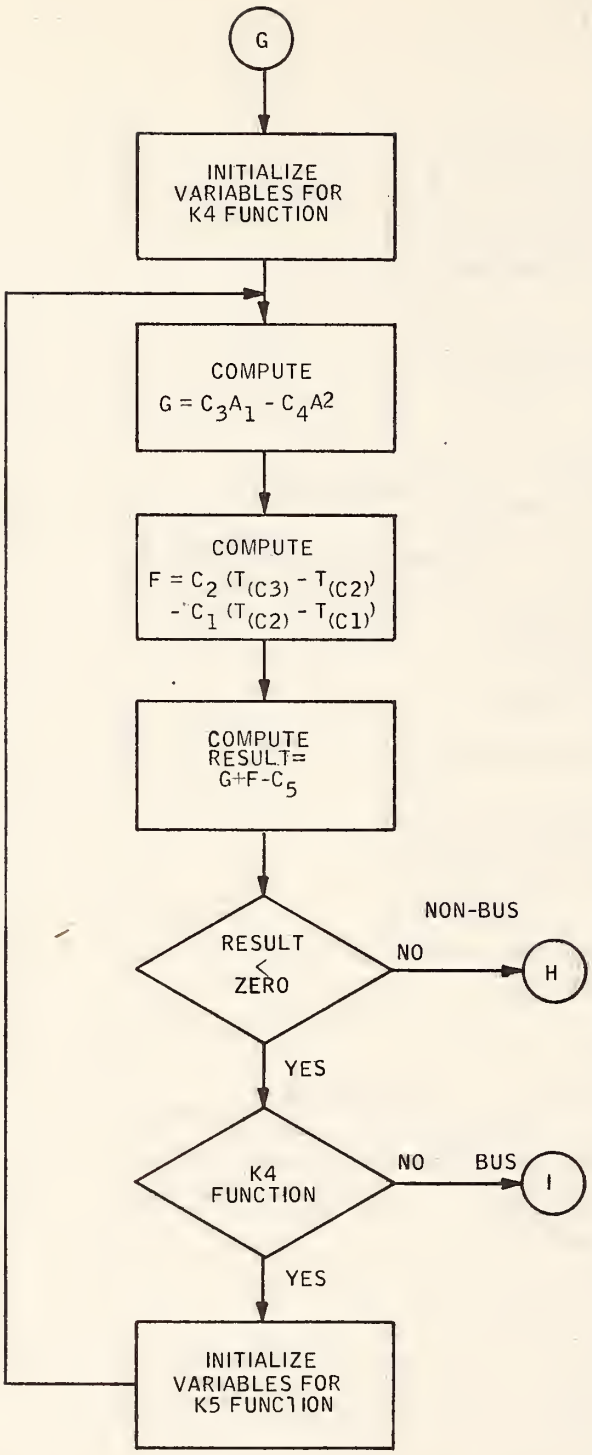


Figure A-2. Flowchart of the FORTRAN IV Vehicle Classifier Vehicle Classification Routine (VCLAS) (Continued)

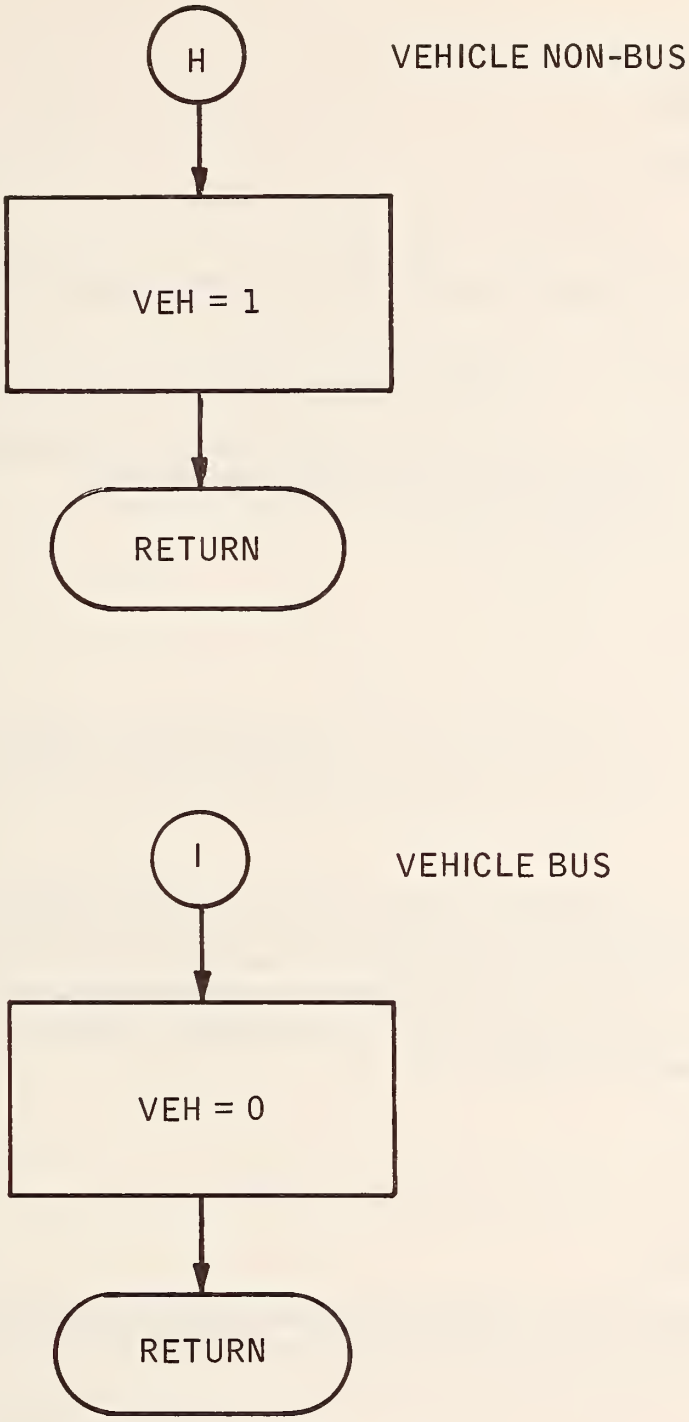


Figure A-2. Flowchart of the FORTRAN IV Vehicle Classifier Vehicle Classification Routine (VCLAS) (Concluded)

3.3 SUBROUTINE ERE

3.3.1 Program Narrative

3.3.1.1 Purpose

This subroutine records the error data supplied by the calling routine.

3.3.1.2 Software Interfaces

This routine is called by the vehicle classification algorithm, subroutine VCLAS. Call arguments for this routine are:

I - Index of error condition

J1 - Error datum value

K - First location in error buffer

Additional data for optional levels of printout are accessed in blank common. No subroutines are called by this routine.

3.3.1.3 Assumptions

The error buffer is assumed to start with the error flag word in which the bits are set to one to indicate the error conditions that have occurred. Following the flags are one pair of error data for each condition: The first is the counter for the number of occurrences, and the second is the error datum value passed to ERE from the calling routine.

3.3.1.4 Solution Approach

The error data are stored in an error buffer.

3.3.1.5 Program Operation

The corresponding error counter is incremented by one and the error datum transferred by the calling routine is stored in the error buffer.

Optional printout is generated based upon the value of the printout control parameter, VCOUT.

3.3.1.6 Model

Not applicable.

3.3.1.7 Data

June, 1975

3.3.1.8 Remarks

None.

3.3.2 Program Work Areas

Refer to the section in the main program documentation for a description of all program work areas.

3.3.3 Program Flowchart

The program flowchart is presented in Figure A-3.

3.3.4 Program Listing

Refer to Appendix D.

3.3.5 Output Data Formats

The test data section of the main program documentation contains examples for all options of printout.

3.3.6 Report Formats

Refer to the test data section of the main program documentation.

3.3.7 Test Data

Refer to the corresponding section of the main program documentation.

3.3.8 Revisions

None.

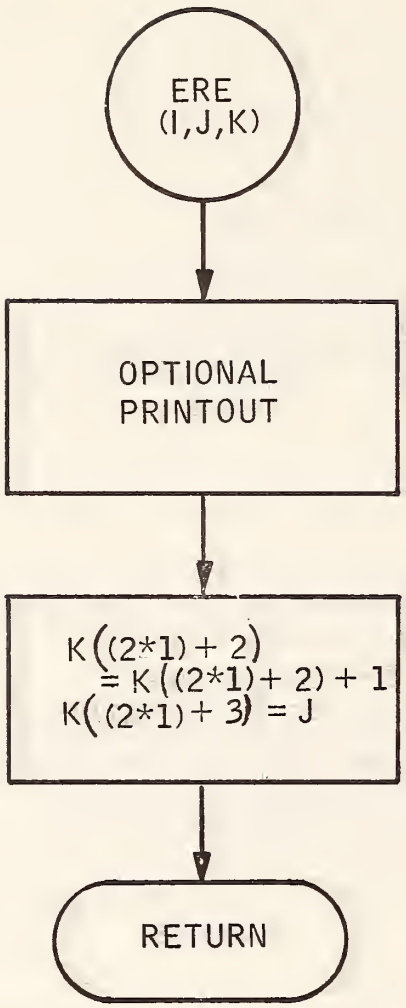


Figure A-3. Flowchart of the FORTRAN IV Vehicle Classifier Error Handling Routine

APPENDIX B

BUS PREEMPT CONTROL UNIT PROGRAM FOR MINNEAPOLIS INSTALLATION

3.0 PROGRAM DOCUMENTATION

3.0.1 Identification

This document describes the computer program identified as the Passive Bus Detector Software for the field test at Thirty-Third and Johnson Street Northeast in Minneapolis, Minnesota. This program is also identified by the acronym, PBD33J. This software is operational in the microcomputer unit (MCU) of the Bus Preempt Control Unit (BPCU) for the demonstration of the concept of bus classification and preemption of the traffic signal timing at the field test site.

This document is written according to the FHWA software documentation standard.

3.0.2 Functional Summary

This program generates the control commands to the traffic signal display incorporating the preemption of the timing plan for clearance of a standard transit bus that is detected and classified. In the absence of preemption, the traffic signals are controlled according to the current background fixed timing plan in synchronization with the loop master controller.

Data for the three timing plans, preemption extended green time, and synchronization with the loop master controller are input to this program from the programming panel on the front of the BPCU.

This program operates in a real time, priority interrupt environment in the MCU and performs all the functions of vehicle signature classification, preemption, traffic signal timing, executive monitoring, scheduling, and self testing.

The program is written in the assembly language of the MCU.

Nomenclature

Abbreviations used in this document are defined at the end of this volume.

Memory Map

The allocation of the MCU memory, the memory address ranges, and memory type are shown in the memory map, Figure B-1. The actual address locations of program components are found in the assembly listing of the entire program in Appendix D.

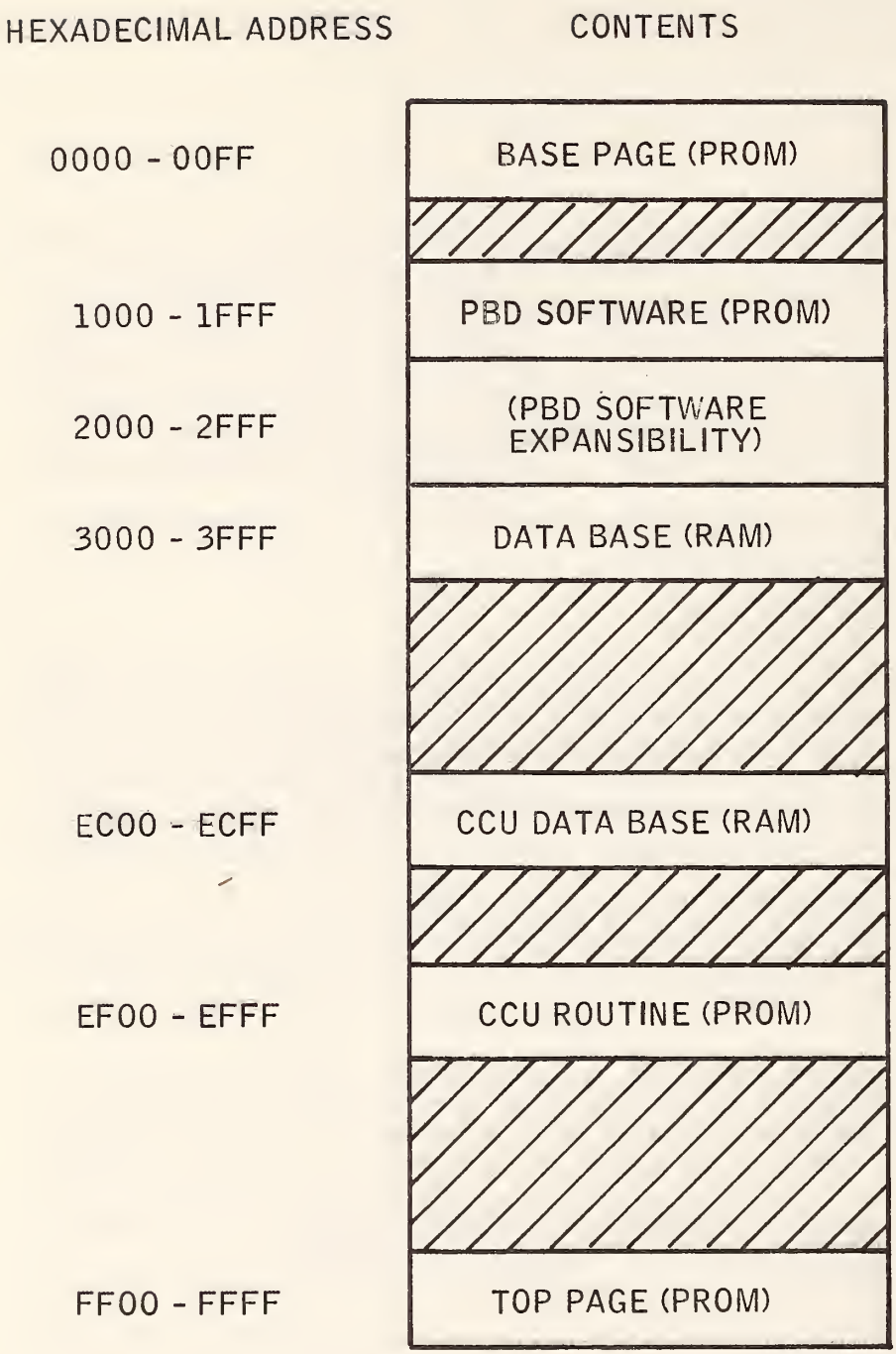


Figure B-1. Memory Map of MCU

Memory Size of Program Components

The memory size of each program component of the passive bus detector software are listed in Table B-1. The spare memory available in the PROM's containing each block of program components are shown along with the address range spanned by each set of PROM's. The base-page and top-page PROM's are 256 x 8 (2 PROM's per 256 x 16 memory range). The PROM's containing the program modules of the PBD software are 512 x 8 (2 PROM's per 512 x 16 memory range).

Program Operation

The overall operation of the Passive Bus Detector Software is shown in Figure B-2.

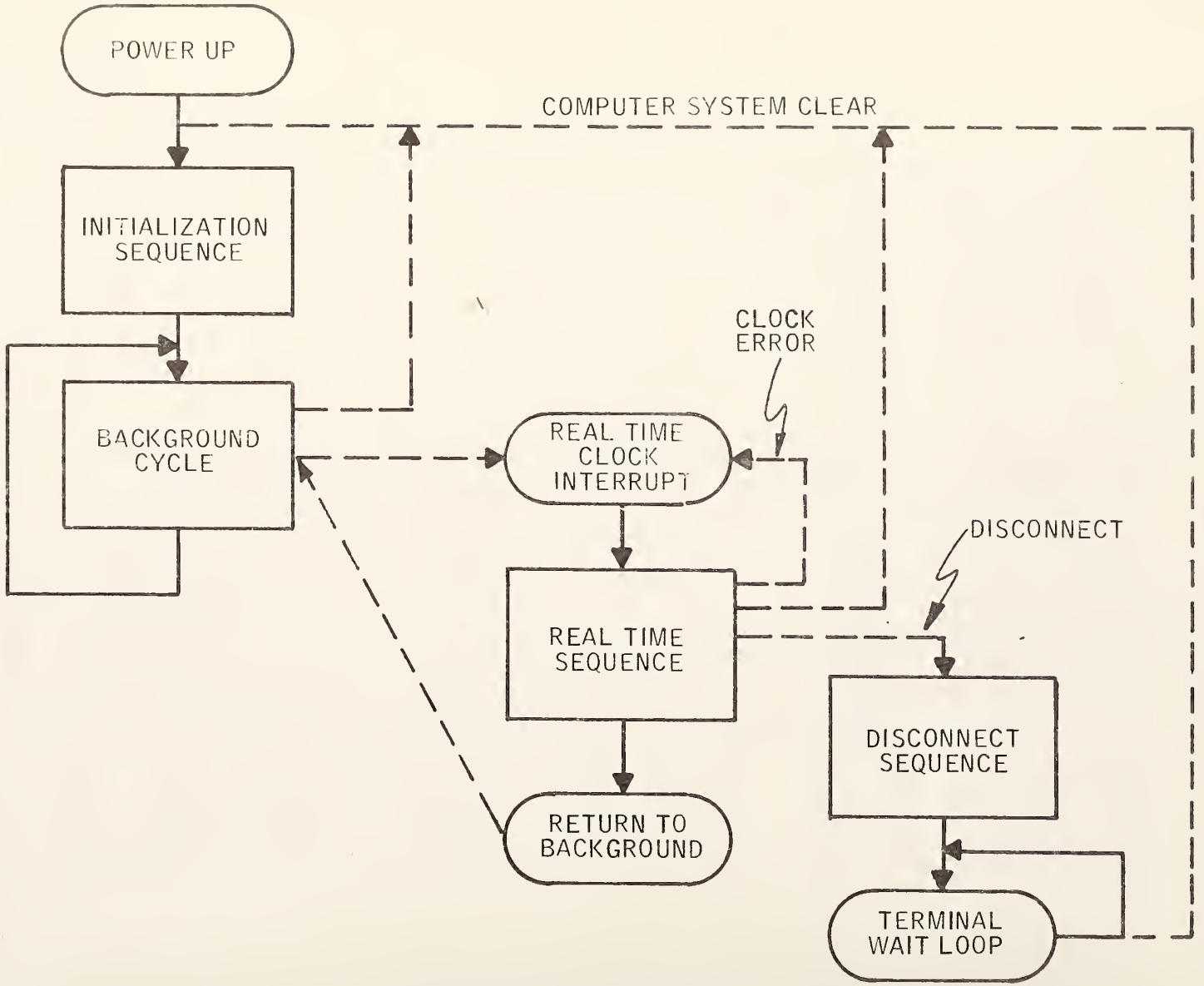


Figure B-2. Passive Bus Detector/Bus Preempt Software Operation

3.1 ASSEMBLY CONSTANTS

The assembly constants section of the source deck defines those constant values that are referenced symbolically throughout the source coding.

3.1.1 Accumulators

The programmable registers or accumulators in the CPU are referenced by the symbols AC0, AC1, AC2, and AC3.

3.1.2 Input/Output Function Codes

These values are referenced in the operand field of input (RIN) and output (ROUT) instructions. The values are limited to 7 bits.

3.1.3 Displacement Constants

These values are used in indexed addressing expressions and in bit displacement operands in the bit manipulation instructions.

3.1.4 Immediate Data Constants

These values are used in the operands of instructions employing immediate data values.

3.1.5 Branch On Condition Codes

These jump condition symbols are used in the operands of the BOC instructions.

3.1.6 Control Flags

These symbols are used for referencing the processor control flags.

3.2 BASE PAGE/DATA BASE CPC

3.2.1 Base Page

The base page is the first 256 locations in memory. This page is directly addressable from anywhere in memory. Some base page locations are reserved for special purposes; the remaining locations are used for full-word constants and linkages to data and program entry points.

3.2.1.1 Reserved Locations

The following locations are reserved in the base page:

X'0000 - Jump through pointer instruction for power up.

X'0001 - Jump through pointer instruction for general interrupt.

X'0002 through X'001F - Reserved for system software use in the software development system environment.

3.2.1.2 Program Entry Points

These data provide the address constants for accessing program entry points.

3.2.1.3 Data Base Pointers

These data provide the address constants for accessing data in the data base.

3.2.1.4 Common Constants

These full-word hexadecimal constants may be referenced by any instruction in memory.

3.2.1.5 Peripheral Device Address Constants

These data provide the constants that are loaded into AC3 in preparation for the input or output instruction which contains the 7-bit function codes. In the execution of the I/O instructions, the contents of AC3 are added to the 7-bit control field in the RIN or ROUT to obtain the peripheral device address and function code.

3.2.2 Data Base

The data base consists of the signature data for the bus classification detectors and the data base for each computer program component.

The information in the glossary of terms is made up of the contents of this section in conjunction with the RAM Data Store/Fetch Matrix of Table B-2 and the assembly listing of the data base in Appendix D.

3.2.2.1 Bus Detector Signature Data

The signature data block for each detector consists of eight values in the following sequence:

1. Total duration of signature
2. Amplitude of first peak
3. Time of first peak
4. Amplitude of second peak
5. Time of second peak
6. Amplitude of last peak
7. Time of last peak
8. End of signature status

Time values have a range of 12 bits with the least significant bit equivalent to 4 milliseconds. Amplitude values have a range of 8 bits and are analyzed only in ratios such that the scaling is not required for classification. Storage is allocated for 32 detectors (8 x 32 = 256 words).

3.2.2.2 Data Input Data Base

BBUF contains the following data:

BDIN - Discrete input data word. (See Table B-3)

BCR1 - Command register 1 status. Feedback of the main street signal display command. (See Table B-4)

BCR2 - Command register 2 status. Feedback of cross street signal display command. (See Table B-5)

BPPD - Discrete switch word for the programmable panel. (See Table B-6)

BPD1 - Thumbwheel switch data for dia 1 on the programmable panel. (See Table B-7)

BPD2 - Same as BPD1 except for dial 2.

BPD3 - Same as BPD1 except for dial 3.

3.2.2.3 Data Output Data Base

XBUF contains the following data:

XSDW - Status display word. (See Table B-8)

XŌCW1 - Output command word 1. Signal display command for main street. (See Table B-9)

XŌCW2 - Output command word 2. Signal display command for cross street. (See Table B-10)

XCIWC - Current command index word. Bits 15 through 8 are the main street and bits 7 through 0 are the cross street command indices, respectively. Refer to the command format table XCFT in routine XSDC.

XCIWP - Previous command index word.

3.2.2.4 Preempt Data Base

PBUF contains the data explained in this subsection.

PIP - Preempt in Progress Flag

Set by Preempt Request Routine (PREQ) when a preempt condition is initialized, PIP contains the detector index number for the clearance detector. PIP is used to determine if a preempt is in progress and to establish a priority scheme in the inspection of vehicle signatures.

PTEM - Preempt Temporary Storage Word

Used for short-term storage of intermediate results.

PTCP - Primary Contractible Interval Address

Preempt Request Routine (PREQ) computes the address of the primary contractible interval (CI_p) from the indices provided by the approach parameters (TECI) and places the address in PTCP for later reference by PREQ.

PTCS - Secondary Contractible Interval Address

Preempt Request Routine (PREQ) computes the address of the secondary contractible interval (CI_s) from the indices provided by the approach parameters (TECI) and places the address in PTCS for later reference by PREQ. If no CI_s can be specified, PTCS is set to zero.

PTEI - Primary Extendible Interval Address

Preempt Request Routine (PREQ) computes the address of the extendible interval (EI) from the indices provided by the approach parameters (TECI) and places the address in PTEI for later reference by PREQ.

PMIF - Preempt Monitor Index Flag

Used as a communication link between Preempt Request Routine (PREQ) and Preempt Termination Routine (PTER), PMIF is a three-word array that contains indices to the extendible interval, primary contractible interval, and secondary contractible interval. These indices are placed in PMIF by PREQ. If an interval occurs in the next cycle, bit 15 is set to one (1). At the end of each cycle, the End of Cycle Routine (EEOC) sets bit 15 of each word to zero. PTER uses these indices to restore any unexpired extension time to subsequent contractible intervals.

PFREQ - Preempt Request Flag

PFREQ is set by the Preempt Request Routine (PREQ) following the modification of the timing plan. When PFREQ is non-zero, the End of Cycle Routine (EEOC) resets the "wait for synchronized start" flag and thus prevents the wait for the synchronized start dwell from disrupting the preempt operation.

PTAVT - Available Time for Interval Restoration

Used as a communication link between Preempt Termination Routine (PTER) and Preempt Contracted Interval Modification Routine (PCIM), PTAVT contains the amount of time available to restore a previously contracted interval.

PTITE - Interval to be Restored

Used as a communication link between Preempt Termination Routine (PTER) and Preempt Contracted Interval Modification (PCIM), PTITE contains an index to the contracted interval that is to be restored. Copied from PMIF by PTER, PTITE is used by PCIM to compute the address for the contracted interval in the timing plan table.

PTPMT - First Location of Master Timing Plan Address

Used as a communication link between Preempt Termination Routine (PTER) and Preempt Contracted Interval Modification (PCIM), PTPMT contains either the address to the first location of the master timing plan for the current dial setting or zero. Set by PTER, if PTPMT is non-zero, an address pointer to the interval that corresponds to the contracted interval (CI) is computed and used to limit expansion of CI to the duration specified in the master timing plan.

PEMAX - Maximum Preempt Extension Parameters

PEMAX is a four-word array that contains:

PEMAX(0) - The maximum extension allowed for the current dial setting.

PEMAX(1) - The maximum extension allowed for the first dial setting.

PEMAX(2) - The maximum extension allowed for the second dial setting.

PEMAX(3) - The maximum extension allowed for the third dial setting.

PEMAX(0) - Is set by Preempt Request Routine (PREQ), which reads the current dial setting (TCDS) and copies the proper word. PREQ uses PEMAX(0) to limit expansion of the extendible interval (EI).

VCDT - Vehicle Detector Index

Used as a communication link between PMON, PREQ, PTER, and VCLAS. VCDT identifies the detector currently under scrutiny and, hence, the approach.

VCBT - Recent Bus Classification Timer

Set by the Vehicle Classification Routine, VCLAS, whenever a vehicle is classified as a bus, a non-zero value in VCBT causes a 1 to be set in bit 6 of the status display word. A positive value in VCBT is decrement by 1 in the Preempt Monitor Routine (PMON).

VCVC - Vehicle Counters

An array 32 words long that is used in vehicle detector failure monitoring. Each word is incremented when the corresponding vehicle detector records a vehicle signature.

VCDB - Vehicle Classifier Data Base

An array 25 words long. The following is copied from the Signature Data Block (SDB):

VDDB(1) - Total duration of signature (x 4 msec)

VCDB(2) - First peak amplitude

VCDB(3) - Time from "turn on" to first peak (x 4 msec)

VCDB(4) - Second peak amplitude

VCDB(5) - Time from "turn on" to second peak (x 4 msec)

VCDB(6) - Last peak amplitude

VCDB(7) - Time from "turn on" to last peak (x 4 msec)

The following is determined during the moving/ stopped test.

VCDB(8) - (VCDB(1) + (VCDB(3) - (VCDB(7))

The following is determined during the stopped vehicle classifier.

VCDB(9) - (G1 coefficient for H1 function) * (VCDB(4))

VCDB(10) - (G2 coefficient for H1 function) * (VCDB(6))

VCDB(11) - (G1 coefficient for H2 function) * (VCDB(4))

VCDB(12) - (G2 coefficient for H2 function) * (VCDB(6))

VCDB(13) - H1 function

VCDB(14) - H2 function

The following is determined in the moving vehicle classifier.

VCDB(15) - (G1 coefficient for K4/K5 function) * (VCDB(4))

VCDB(16) - (G2 coefficient for K4/K5 function) * (VCDB(6))

VCDB(17) - (VCDB(16) - (VCDB(17)/ (VCDB(2)))

VCDB(18) - (F1 coefficient for K4/K5 function) * (VCDB(5) - (VCDB(3))

VCDB(19) - (F2 coefficient for K4/K5 function) * (VCDB(7) - (VCDB(5))

VCDB(20) - (VCDB(19) - (VCDB(18)/(VCDB(8)))

VCDB(21) - K4 function

VCDB(22) - K5 function

VCDB(23-25) - Not used.

VCLOG - Vehicle Classifier Vehicle Log

An array 15 words long.

VCLOG(1) - Number of signatures rejected as noise (total duration less than 128 msec).

VCLOG(2) - Number of signatures that failed gross features test and are thus classified as non-buses (fewer than three peaks).

VCLOG(3) - Number of signatures classified by stopped vehicle classifier.

VCLOG(4) - Number of signatures classified by moving vehicle classifier.

VCLOG(5) - Number of signatures that passed gross features test and were classified as non-buses.

VCLOG(6) - Number of signatures classified as buses.

VCLOG(7-15) - Not used.

3.2.2.5 Timing Plan Data Base

TBUF contains the data explained in this subsection.

TCIWC - This is the current command index word buffer. The current command index word is selected from the command index word table, TCIW, by TPEX and is stored in TCIWC for use by the data output routine.

TCDS - This is the current dial selection indicator. It is tested by the TNCY routine at the end of a cycle. It is compared with the new dial selection, which is indicated by the dial select

bits in the interval dial select word ED23. If different, the TNCY copies the appropriate new master timing plan to the current and next timing plan buffers via TCOPY. The required new value (1, 2, or 3) is then stored in TCDS via TNCY.

TIN - This is the current timing interval index. TIN is initialized to the index of the transfer hold interval. It is incremented via TNXT at the end of each timing plan interval. TIN is reset to zero (0) via TNXT when it points to a negative interval length in the current timing plan buffer. TIN is also referenced by routines in the Executive CPC and the Preempt CPC.

TINT - This is the current interval timer. TINT contains the time, in 0.1-second real time clock intervals, remaining in the current timing plan interval. It is initialized to the duration of the next timing interval via TNXT upon completion of the previous interval. It is decremented via TPEX until the minimum value of zero (0) is reached. A zero value indicates the end of the current interval and causes a jump to TNXT to be executed. TINT may be modified by the Preempt CPC in the case of a preempt request or preempt termination during an extendible interval.

TFST - This is the "wait for synchronized start" flag. It is set to a value of one (1) by the end of cycle routine (EEOC) in the Executive CPC. It can be reset to zero (0) via EPEOC in the case of a preempt in progress at end of cycle. If TFST has a value of one (1), timing is delayed until the wait for synchronized start dwell timer reaches a maximum of 24 percent of the cycle. Bit 15 of TFST is set by TSST when the timing plan is released to start timing the first interval in the MSG sequence. Bit 15 communicates the event of timing plan release via the data base.

TDWL - This is the maximum "wait for synchronized start" dwell buffer. TDWL is the first location of a three-element buffer. The first location contains the maximum synchronized start dwell, in 0.1-second real time clock intervals, for dial 1; the next location contains the maximum dwell for dial 2; and the third location contains the maximum dwell for dial 3. The maximum dwell is 24 percent of the current cycle length. It is computed by EPP1, EPP2, and EPP3 via EM24 and put into the data base by EPUP.

TPMI - This is the beginning location of the read/write buffer for the dial 1 master timing plan table. The buffer is presently 20 locations, and for a timing plan with n intervals would be allocated as follows:

Relative Location	Contents
1	Length in tenths of seconds of the first interval in the timing plan.
2	Length in tenths of a second of the second interval in the timing plan.
.	.
.	.
.	.
.	.
.	.
n	Length in tenths of a second of the last interval in the timing plan.
n+1	A negative number used to signal the end of the timing plan.

Any additional locations in the buffer are blank.

The contents of the TPM1 buffers are initialized by ERSI to the corresponding contents of the skeleton timing plan tables from the Timing Plan Data Tables portion of the Timing Plan CPC. Two adjustable intervals, whose location relative to TPM1 are specified by the contents of TAI1 and TAI2 from the Timing Plan Data Tables, can be modified by the MSG split selection on the Programmable Panel via EPUP.

TPM2 - This is the beginning location of the read/write buffer for the dial 2 master timing plan table. The buffer is the same length as for the dial 1 master and is allocated, initialized, and modified in the same manner.

TPM3 - This is the beginning location of the read/write buffer for the dial 3 master timing plan table. The buffer is the same length as for the dial 1 master and is allocated, initialized, and modified in the same manner.

TNTP - This is the beginning location of the read/write buffer for the next cycle planning plan. The contents of this buffer is refreshed at the end of each cycle with the contents of the TPM1, TPM2, or TPM3 buffers, whichever is consistent with the dial selection. The contents of this buffer can also be modified by the preempt routines. The TNTP buffer is the same length as the TPM buffer.

TCTP - This is the beginning location of the read/write buffer for the current cycle timing plan. The contents of this buffer is refreshed at the end of each timing cycle with the contents of the TNTP buffer (before it is refreshed) or, in the case of a dial change, with the contents of the appropriate master buffer. The contents of the TCTP buffer can also be modified by the preempt routines during execution of the timing plan. The TCTP buffer is the same length as the TPM1 buffer.

3.2.2.6 Executive Data Base

EBUF contains the data explained in this subsection.

ERTCF - Real time computation sequence compute flag. (=1 RTC sequence in process; =0 RTC sequence completed)

ETST - Synchronization dwell timer that is incremented by a timing function. The timing plan routine dwells at MSG for synchronization in the absence of a preempt until the start pulse EPST occurs or until this timer reaches the maximum dwell time.

ETCY - Current measured cycle time including all modifications.

ETCY+1 - Previous maximum measured cycle time.

ETCY+2 - Previous minimum measured cycle time.

EPST - Start pulse. (=0 no pulse; ≠0, pulse) Tested by TSST during synchronization dwell. Set and reset by ESYN.

EPDT - Not used.

ECMF - Command monitoring temporary error flag (=0, no error; ≠0, error) Used to communicate command feedback error status from ECOM to XSDC.

ESYD - Synchronization data buffer. Six values for each of the three dials in the following sequence:

cycle length	reset 3
reset 1	end of start pulse
reset 2	synch dial timer

ER123 - Reset select data after processing through the software filter.

ERN0 - Three past values of the reset select data.

ED23 - Dial select data after processing through the software filter.

EDN0 - Three past values of the dial select data.

EDMIT - Detector monitoring interval timer. Incremented by ETIM up to the value EDETI.

EDETT - Temporary storage for EDET during the error recording sequences.

Each of the save areas below contains the computational environment as stored at that event. The data appears in the following sequence in ESAV, CSAV, and EDSAV:

AC0, AC1, AC2, AC3

RALU flags

CPU stack contents

Select flag state

ISV contains only the accumulators and the stack.

ESAV - Save area for real time clock interrupts

CSAV - Save area for catch-on-the-fly operation

ISV - Stack full save area

EDSAV - Disconnect save area

CTST - Optional uses during software development

EPD1, EPD2, EPD3 - Converted integer values for the programmable panel thumbwheel data for the three timing plans. The data is allocated in the following sequence (0.1-second units):

First adjustable interval length

Second adjustable interval length

Maximum extended green

Cycle length

Reset 1

Reset 2

Reset 3

Start pulse duration

Maximum synchronization dwell at MSG

INT2 (not used)

EPFT - Sum of the fixed intervals in each of the three timing plans.

EP1E, EP2E, EP3E - Error flags for the programmable panel thumbwheel switch data for the three timing plans.

EPDVE - Data validity error flag for all programmable panel thumbwheel switch data.

EPUF - Programmable panel interval data update flag. (≠0 update data base at MSG; =0 no update)

EPCF - Computation flag in EPPM to control the processing of programmable panel data over several RTC intervals.

= 0, no processing

= -1, input thumbwheel switch data

= 1, process plan 1 data

= 2, process plan 2 data

= 3, process plan 3 data

= 4, completion of sequence

EPIR - Previous state of input request switch on the programmable panel.

SSAD - Start address for RAM test.

SEAD - End address for RAM test.

3.2.2.7 Error Buffers

The buffers for error data are identified as BERR, XERR, PERR, TERR, EERR, EPEF1, EPEF2, EPEF3. These buffers are headed by an error flag word. The bits in the error flag word are assigned individually to error conditions. For each error condition, there are two full words in the error buffer: the first word is the counter for the number of times this error has occurred, and the second word is the error value related to the error condition.

Following these data are the detector undercounting error data for up to 32 detectors.

BERR - Data Input CPC error buffer (no error conditions assigned).

XERR - Data Output CPC error buffer.

XERRF - Error flags

Bit 0, invalid command index for main street

Bit 1, invalid command index for cross street

PERR - Preempt CPC error buffer

PERRF - Error flags

Bit 0 - Arithmetic overflow in VCLAS.

Bit 1 - Detector index out of range in VCLAS.

Bit 2 - Amplitude datum out of range.

Bit 3 - Time-datum out of range.

Bit 4 - Data out of sequence.

Bit 5 - Noise.

Bit 6 - New signature started during classification data copy operation.

Bit 7 - Signature lost during copy operation.

Bit 8 - Detector index out of range in PREQ.

Bit 9 - Preempt request started for downstream detector.

TERR - Timing plan CPC error buffer

TERRF - Error flag word. Individual bits are used as flags. The set condition (bit = 1) indicates errors have occurred as follows:

Bit Ø - Dial select error. Bit Ø is set if bits 4 and 5 of ED23 are both 1 during the test for a dial change in TNCY. It is a non-fatal error. Default logic causes the dial selection to remain the same as for the previous cycle.

Bit 1 - Interval index out of limits error. Bit 1 is set by the TNXT routine if the current interval index (TIN) is less than zero or greater than a maximum value (TIMX) stored in ROM in the Timing Plan Data Table. It is a fatal error and causes TNXT to execute a jump to the autodisconnect routine (EADC).

Bit 2-15 - Unused.

3.3 EXECUTIVE CPC

This computer program component contains the executive functions for the PBD software. These functions include initialization, interrupt processing, monitoring, timing, scheduling, self testing, synchronization, status, programming panel, error recording, and autodisconnect.

3.3.1 Restart Initialization Routine (ERSI)

3.3.1.1 Program Narrative

3.3.1.1.1 Purpose

This routine performs the initialization sequence at power up of the MCU.

3.3.1.1.2 Software Interfaces

This routine is entered via a jump through pointer at memory location X'0000, which is reached by the jump direct instruction at X'FFFE. At initial power up, the PC is forced to X'FFFE to start this transfer sequence. This routine calls the following routines:

ECCU - CCU initialization

EMAD - RAM test

EPAN - Programmable panel

BRTC - Discrete data input

TDIAL - Timing plan initialization

XSDC - Output command formatting

The background computation cycle is entered by loading the entry point on the stack and executing a return instruction.

Table B-2 is a list of the data fetched and the data stored by this routine.

3.3.1.1.3 Assumptions

This program is entered only at the power up of the MCU.

3.3.1.1.4 Solution Approach

This routine controls the sequence that initializes peripheral devices with appropriate control pulses, initializes the CCU, performs the RAM test, clears processor status flags and stack, initializes the free RAM entry point, zeroes the RAM data base, initializes the timing plan, programmable panel and output command word, starts the real time clock, and initiates the background computation cycle.

A jump condition is tested just prior to starting the RTC to provide a pseudo halt, which is controlled by an external switch. The status of the MCU following initialization can then be examined with the CCU.

3.3.1.1.5 Program Operation

The stack full condition is tested immediately to preclude an error on a subsequent subroutine call. The stack full condition results in an autodisconnect. Next, the RTC is assured to be stopped, the RTC interrupt pulse is reset, and the general interrupt line is pulsed. ECCU is called to initialize the CCU RAM and servicing routine. Status flag 4 is cleared and the initial RAM test is performed by calling EMAD. Status flags 5, 6, and 7 are reserved for expansibility. The first location in the RAM that is free for the insertion of patch code and is accessed via routine COTF is initialized with a standard return instruction. A reset pulse is issued to the conflict monitor, the CPU stack is initialized for the background programs, the data base is cleared to zeroes and then the following data are initialized:

Master timing plan tables

Minimum measured cycle length

Programmable panel data

Discrete input word

Timing plan duration tables

Current interval index

Current interval timer

Output command words.

The pseudo halt sense switch is tested and the program will loop at this instruction while the switch remains on. The RTC is then started. The background computations are initiated and interrupts are enabled by the final return instruction.

3.3.1.1.6 Model

Not applicable.

3.3.1.1.7 Date: July 1975

3.3.1.1.8 Remarks

None.

3.3.1.2 Program Work Areas

The initialization sequence controlled by this routine affects every location in RAM except the CCU RAM. The RAM address test affects every location except the data base and CCU RAM. The entire data base including the bus detector signature data is cleared to zero. Data base locations are then initialized as appropriate.

3.3.1.3 Program Flowchart

The program flowchart is presented in Figure B-3.

3.3.1.4 Program Listing

The program listing is presented in Appendix D.

3.3.1.5 Output Data Formats

The output control pulses issued by this routine are completely defined by the peripheral device address and function code. The contents of AC0 are not used.

3.3.1.6 Report Formats

Not applicable.

3.3.1.7 Test Data

To be determined.

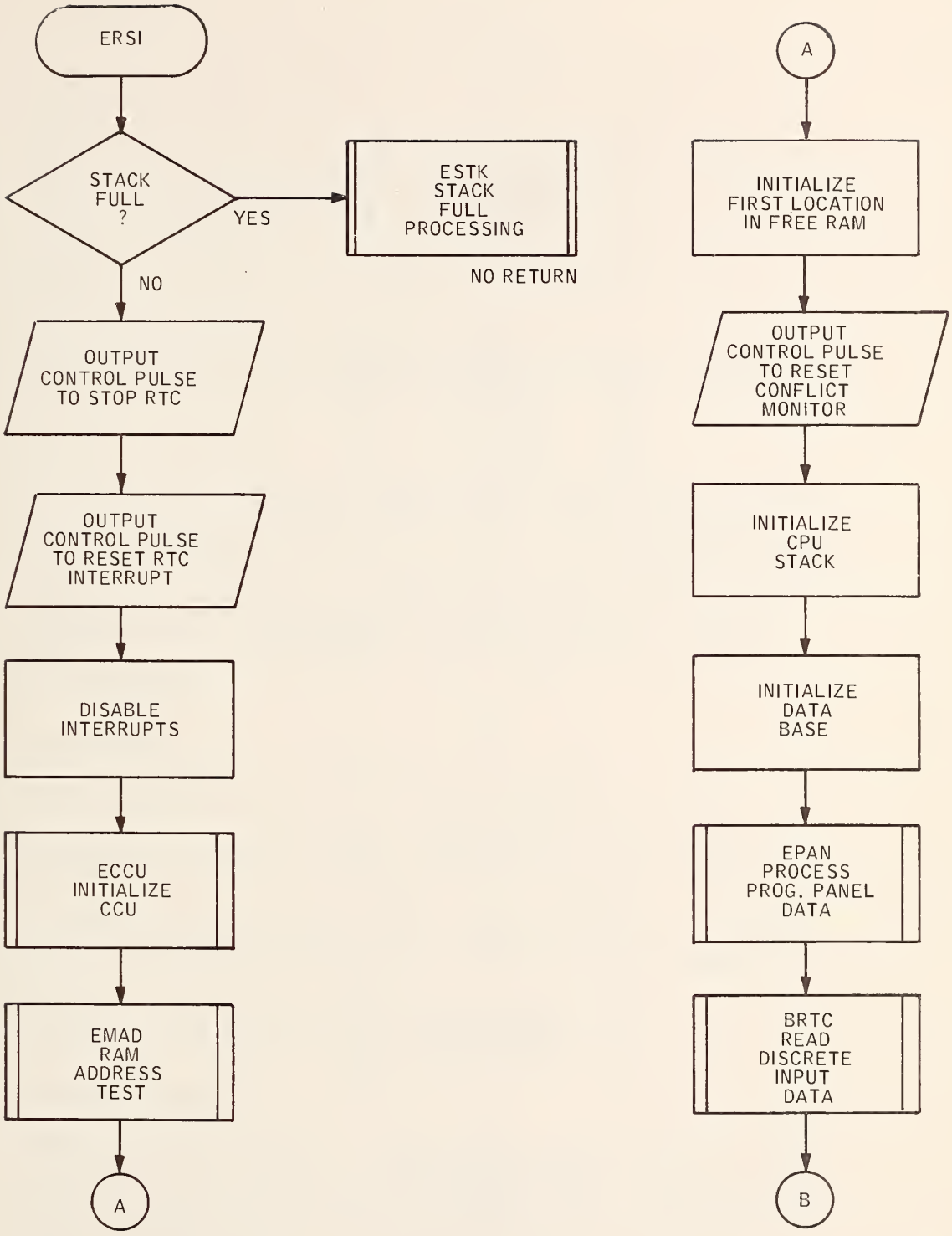


Figure B-3. Restart Initialization Routine (Executive CPC)

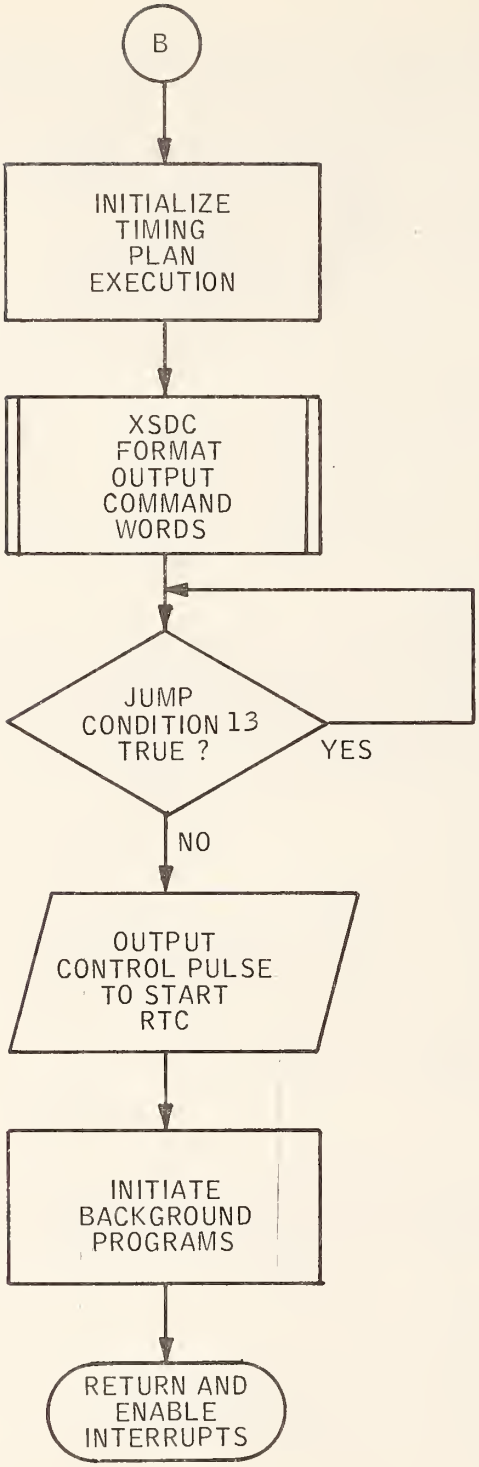


Figure B-3. Restart Initialization Routine (Executive CPC) (Concluded)

3.3.1.8 Revisions

None.

3.3.2 General Interrupt Servicing Routine (EGIS)

3.3.2.1 Program Narrative

3.3.2.1.1 Purpose

This routine services interrupts on the general interrupt line.

3.3.2.1.2 Software Interfaces

This routine is entered via the jump through pointer instruction at memory location X'0001 as the general interrupt forces the program counter to X'0001. In servicing the RTC interrupt, the executive scheduling routine, ESCH, is called. In the event of critical conditions which preclude further operation, the autodisconnect routine, EADC, is called. Table B-2 is a listing of the data fetched and the data stored by this routine.

3.3.2.1.3 Assumptions

This routine is entered only at the general interrupt.

In addition to the stack full condition, the RTC is the only source of an interrupt on the general interrupt line.

The stack full processing logic at ESTK must be within the PC relative addressing range of the stack full test in ERSI.

The RTC computation flag is assumed to be in the first location in the executive data base at EBUF.

3.3.2.1.4 Solution Approach

The stack full condition is tested for and serviced. A data flag is used to diagnose a clock error. Two words of all ones are loaded on the stack as a precaution against a stack full condition; if this condition exists, data is lost from the stack. A stack full or clock error condition results in autodisconnect.

3.3.2.1.5 Program Operation

The stack full condition is tested for immediately and, if present, cleared as quickly as possible, followed by the saving of the remainder of the computational environment and the stopping of the RTC. The autodisconnect routine is then entered. In the absence of the stack full condition, the computational environment is saved and the RTC interrupt servicing at ERTC is entered. An RTC computation flag, ERTCF, is set and reset to indicate the durations of the RTC sequence for clock error diagnosis. If this flag is set when the RTC interrupt is serviced, the previous RTC sequence is assumed to be incompletely processed and the clock error processing is initiated. In the absence of the clock error, the RTC compute flag is set, the stack is initialized for the RTC sequence and the executive scheduling routine, ESCH, is called. Upon return, the stack is cleared, the computation flag is reset, the computational environment is restored, and the final return restarts the interrupted background computations.

3.3.2.1.6 Model

Not applicable.

3.3.2.1.7 Date: July 1975

3.3.2.1.8 Remarks

None.

3.3.2.2 Program Work Areas

The computational environment at the interrupt is saved in the RTC save area at ESAV or the stack full save area at ISV, as appropriate.

Clock error data are stored in the executive error buffer, EERR. The RTC computation flag is ERTCF in the executive data base.

3.3.2.3 Program Flowchart

The program flowchart is presented in Figure B-4.

3.3.2.4 Program Listing

The program listing is presented in Appendix D.

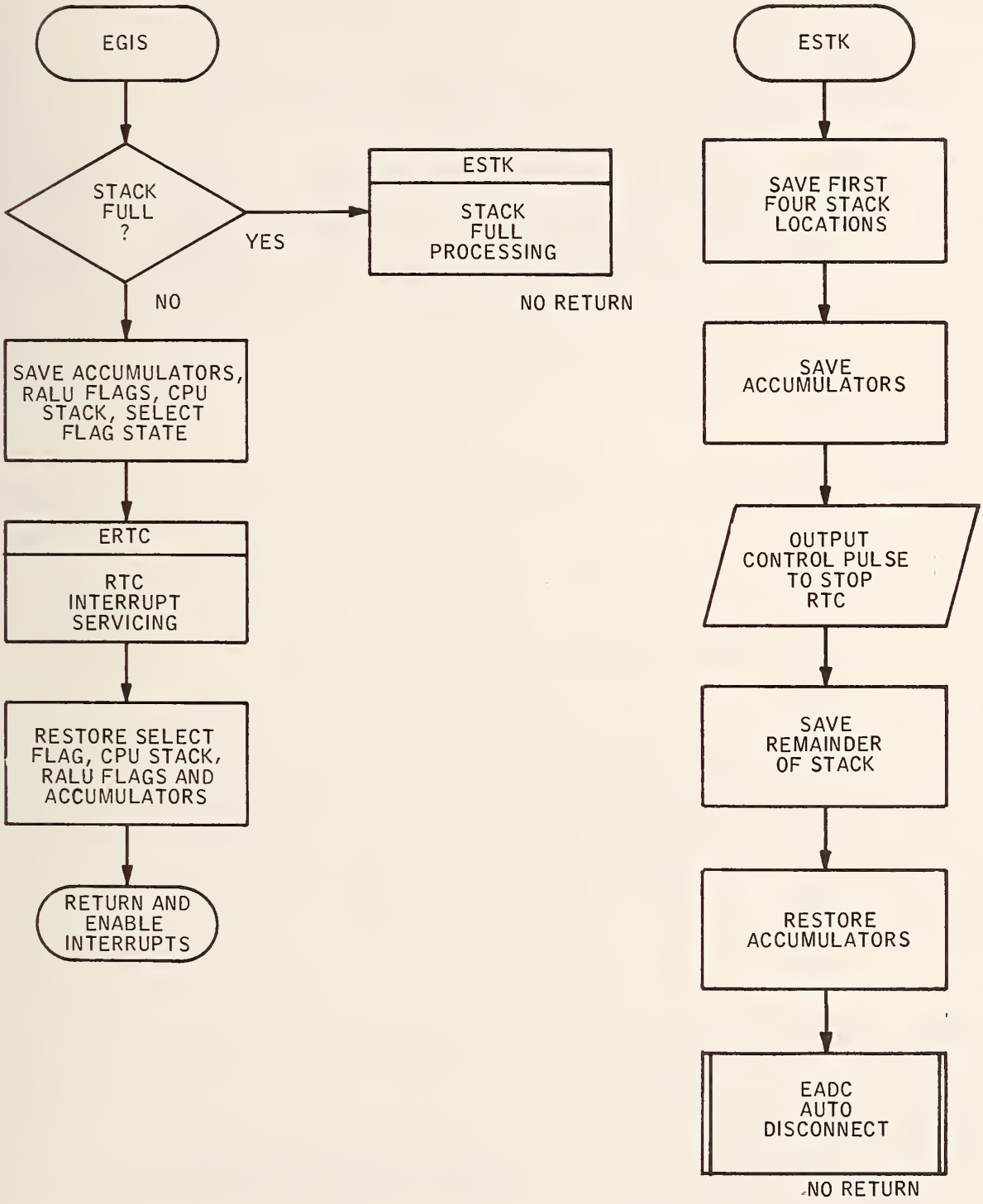


Figure B-4. General Interrupt Servicing Routine (Executive CPC)

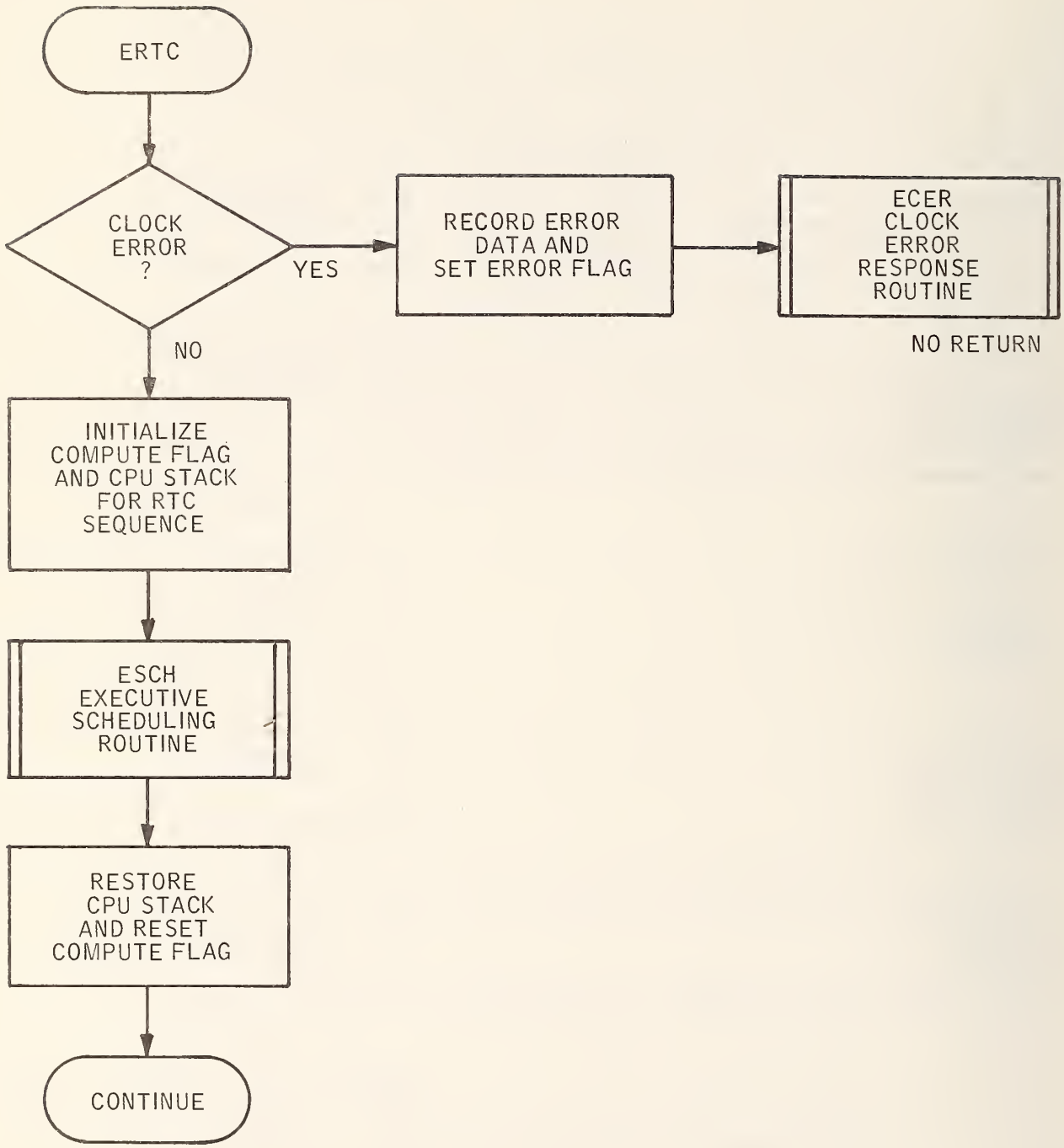


Figure B-4. General Interrupt Servicing Routine (Executive CPC) (Concluded)

3.3.2.5 Input/Output Data Formats

The output control pulse issued by this routine to stop the RTC is completely defined by the peripheral device address and function code. The data in AC0 is not used.

3.3.2.6 Report Formats

Not applicable.

3.3.2.7 Test Data

To be determined.

3.3.2.8 Revisions

None.

3.3.3 Background Program Monitor Routine (EBKD)

3.3.3.1 Program Narrative

3.3.3.1.1 Purpose

This routine controls the background computation cycle.

3.3.3.1.2 Software Interfaces

This routine is initiated by ERSI at the end of the initialization sequence. This routine calls SMAD to perform the RAM test during real time operation. This routine continues to cycle through the background sequence until an error occurs, at which time, this routine enters a terminal wait loop. The error condition is communicated to the disconnect monitor routine via a RALU status flag.

3.3.3.1.3 Assumptions

The RAM test is made over the unused memory from DEND, end of the data base, to X'3BFF just prior to the free RAM available for on-line patch code. Refer to routine COTF.

Status flag 4 is assumed to be cleared prior to initiation of this routine and is set only by routine SMAD when a RAM address error is detected.

3.3.3.1.4 Solution Approach

The status flag is tested upon return from SMAD at the no error return as a precaution. If the status flag is set or if SMAD makes the error return, this routine enters a terminal wait loop to preserve the error data in AC2 and AC3 until the disconnect monitor in the RTC sequence can sense the error condition and initiate the autodisconnect.

3.3.3.1.5 Program Operation

The start and end address of the RAM test are stored in the data base. SMAD is called. Errors tests are made upon return. An error results in the terminal wait loop. No error results in the initiation of the background cycle again.

3.3.3.1.6 Model

Not applicable.

3.3.3.1.7 Date: September 1975

3.3.3.1.8 Remarks

None.

3.3.3.2 Program Work Areas

The start and end addresses for the RAM test are stored in the data base locations SSAD and SEAD, respectively. At the error return from SMAD, AC2 contains erroneous contents at the address in AC3.

3.3.3.3 Program Flowchart

The program flowchart is presented in Figure B-5.

3.3.3.4 Program Listing

The program listing is presented in Appendix D.

3.3.3.5 Input/Output Data Formats

Not applicable.

3.3.3.6 Report Formats

Not applicable.

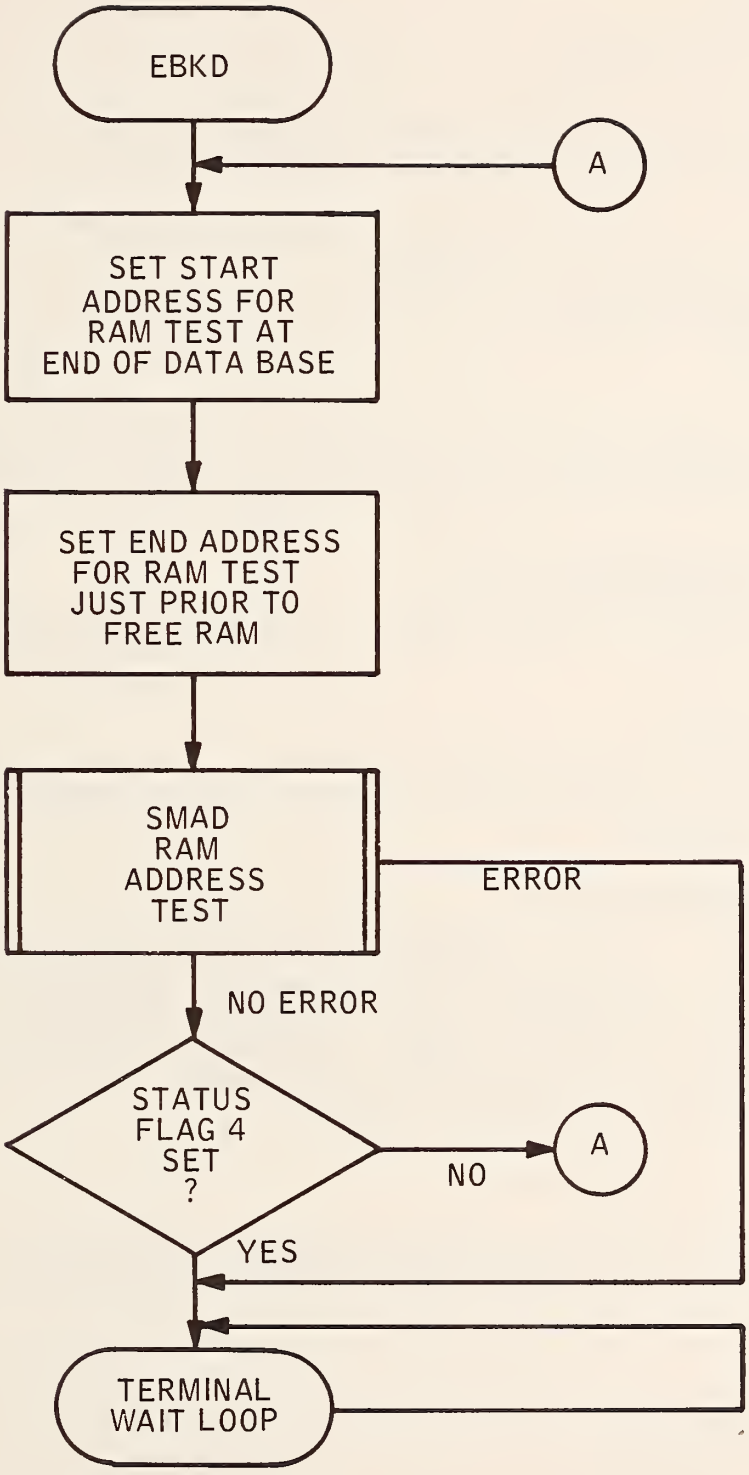


Figure B-5. Background Program Monitor Routine (Executive CPC)

3.3.3.7 Test Data

To be determined.

3.3.3.8 Revisions

None.

3.3.4 Clock Error Routine (ECER)

3.3.4.1 Program Narrative

3.3.4.1.1 Purpose

This routine is executed if the RTC computation sequence is not completed before the next RTC interrupt. This program records the error and initiates an autodisconnect.

3.3.4.1.2 Software Interfaces

This routine is entered only from the RTC interrupt servicing routine ERTC.

3.3.4.1.3 Assumptions

The RTC computation flag is set to 1 during execution of the RTC sequence.

3.3.4.1.4 Solution Approach

The RTC computation flag is set just prior to the call to the executive scheduling routine for the RTC sequence, ESCH, and reset upon return. At the beginning of the RTC interrupt servicing, this flag is tested. If the flag is non-zero, the previous RTC computation sequence was not completed and the clock error routine is entered.

3.3.4.1.5 Program Operation

The error data is stored in the executive error buffer, EERR, and the autodisconnect routine, EADC, is entered.

3.3.4.1.6 Model

Not applicable.

3.3.4.1.7 Date: July 1974

3.3.4.1.8 Remarks

None.

3.3.4.2 Program Work Areas

The RTC compute flag is ERTCF and error data is assigned to error condition 1 in the executive error buffer, EERR.

3.3.4.3 Program Flowchart

The program flowchart is presented in Figure B-6.

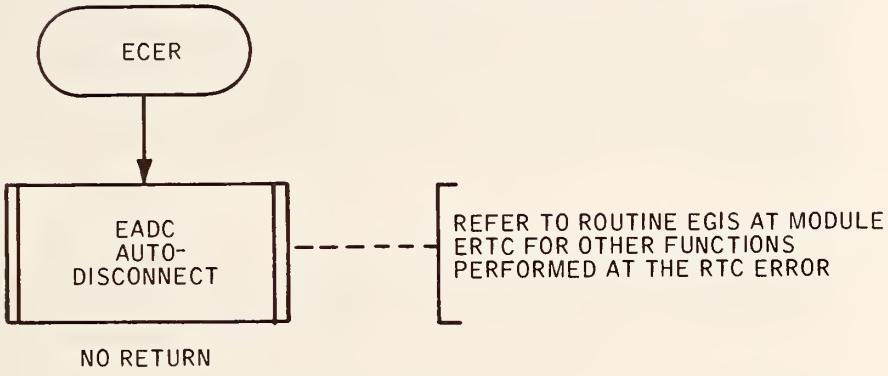


Figure B-6. Clock Error Routine (Executive CPC)

3.3.4.4 Program Listing

The program listing is presented in Appendix D.

3.3.4.5 Input/Output Data Formats

Not applicable.

3.3.4.6 Report Formats

Not applicable.

3.3.4.7 Test Data

To be determined.

3.3.4.8 Revisions

None.

3.3.5 Scheduling Routine (ESCH)

3.3.5.1 Program Narrative

3.3.5.1.1 Purpose

This routine calls the primary subroutine to perform each function in the real time control sequence initiated at each real time clock interrupt.

3.3.5.1.2 Software Interfaces

This routine is called only by the real time clock interrupt servicing routine portion of the general interrupt servicing routine, EGIS. This routine calls the following routines in the RTC sequence: XRTC, BRTC, EDIS, ECOM, ESYN, EMIC, ETIM, EDET, PMON, TPEX, XSDC, ESDW, and EPPM. Prior to this sequence, the catch-on-the-fly routine, COTF, is called conditionally upon a jump condition set by an external switch.

3.3.5.1.3 Assumptions

The output control pulse to reset the RTC interrupt is completed before the next instruction.

3.3.5.1.4 Solution Approach

This routine assures that interrupts are enabled during the RTC sequence such that a clock error may be detected. Capability is provided for insertion of patch code prior to the RTC sequence using a catch-on-the-fly routine that transfers control to a reserved, programmable portion of read/write memory. In the software development environment of read/write memory for this program, the code may be patched to include a loop that causes a delay time following the RTC sequence. This delay may be used to time the execution of the RTC sequence by increasing the delay until a clock error is detected.

3.3.5.1.5 Program Operation

The RTC interrupt pulse is reset and general interrupts are enabled. The jump condition is tested for the conditional call to COTF. The RTC sequence is then executed. If the program is patched in a read/write memory environment, the delay loop is then executed. The delay time depends on the loop counter, which is a local data item in this routine.

3.3.5.1.6 Model

Not applicable.

3.3.5.1.7 Date: July 1975

3.3.5.1.8 Remarks

None.

3.3.5.2 Program Work Areas

This routine sets the interrupt enable flag for the CPU.

3.3.5.3 Program Flowchart

The program flowchart is presented in Figure B-7.

3.3.5.4 Program Listing

The program listing is presented in Appendix D.

3.3.5.5 Output Data Format

The output control pulse to reset the RTC interrupt pulse is completely defined by the function code and peripheral device address. The data in AC0 is not used.

3.3.5.6 Report Formats

Not applicable.

3.3.5.7 Test Data

To be determined.

3.3.5.8 Revisions

None.

3.3.6 Disconnect Monitor Routine (EDIS)

This routine will call the audodisconnect routine if the CPU status flags 4, 5, 6, or 7 are set, indicating errors in the RAM address test and three conditions not yet assigned, respectively. This routine will also initiate the autodisconnect if the external disconnect bit is set in the discrete input word. See Figure B-8.

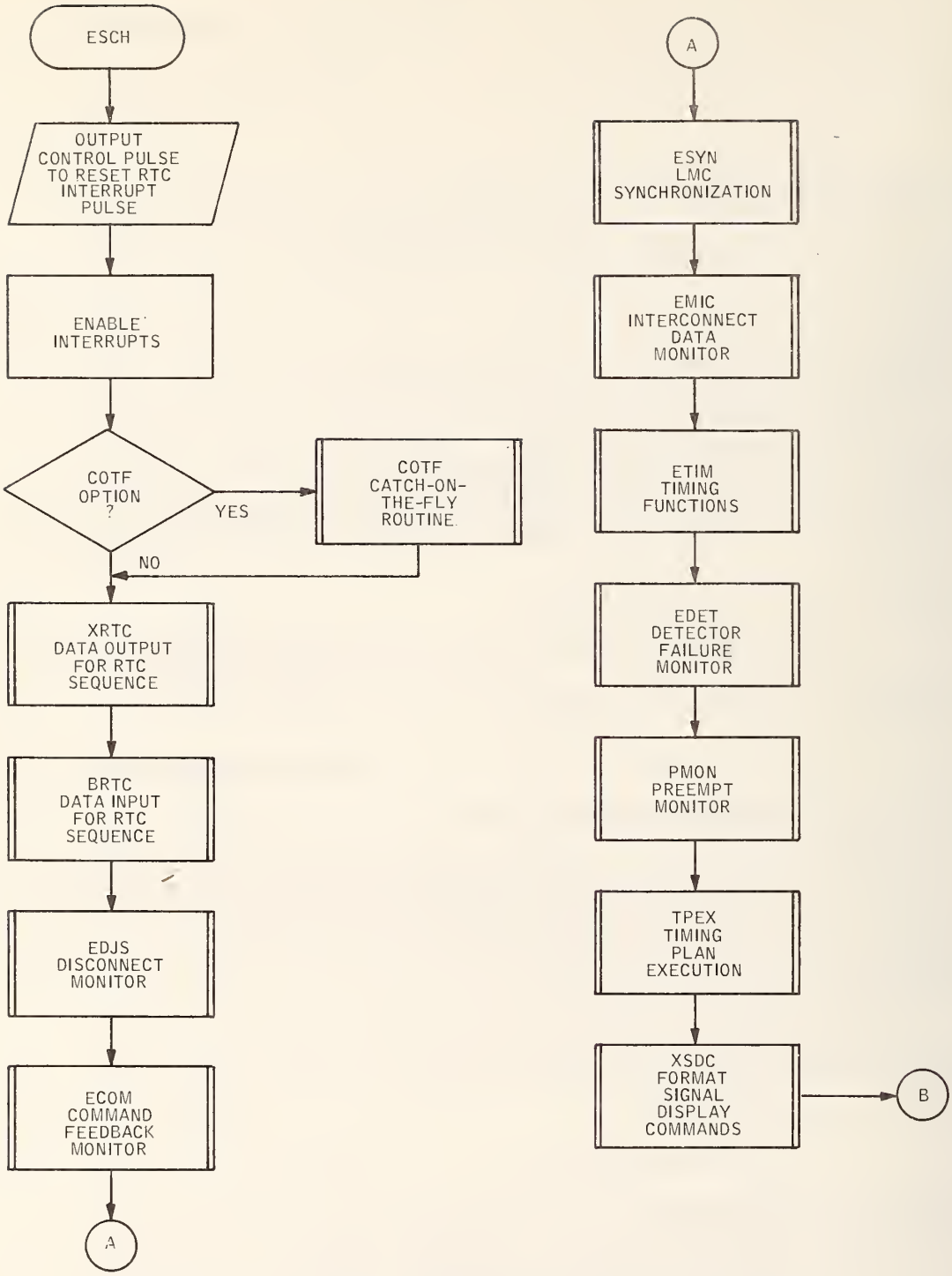


Figure B-7. Scheduling Routine (Executive CPC)

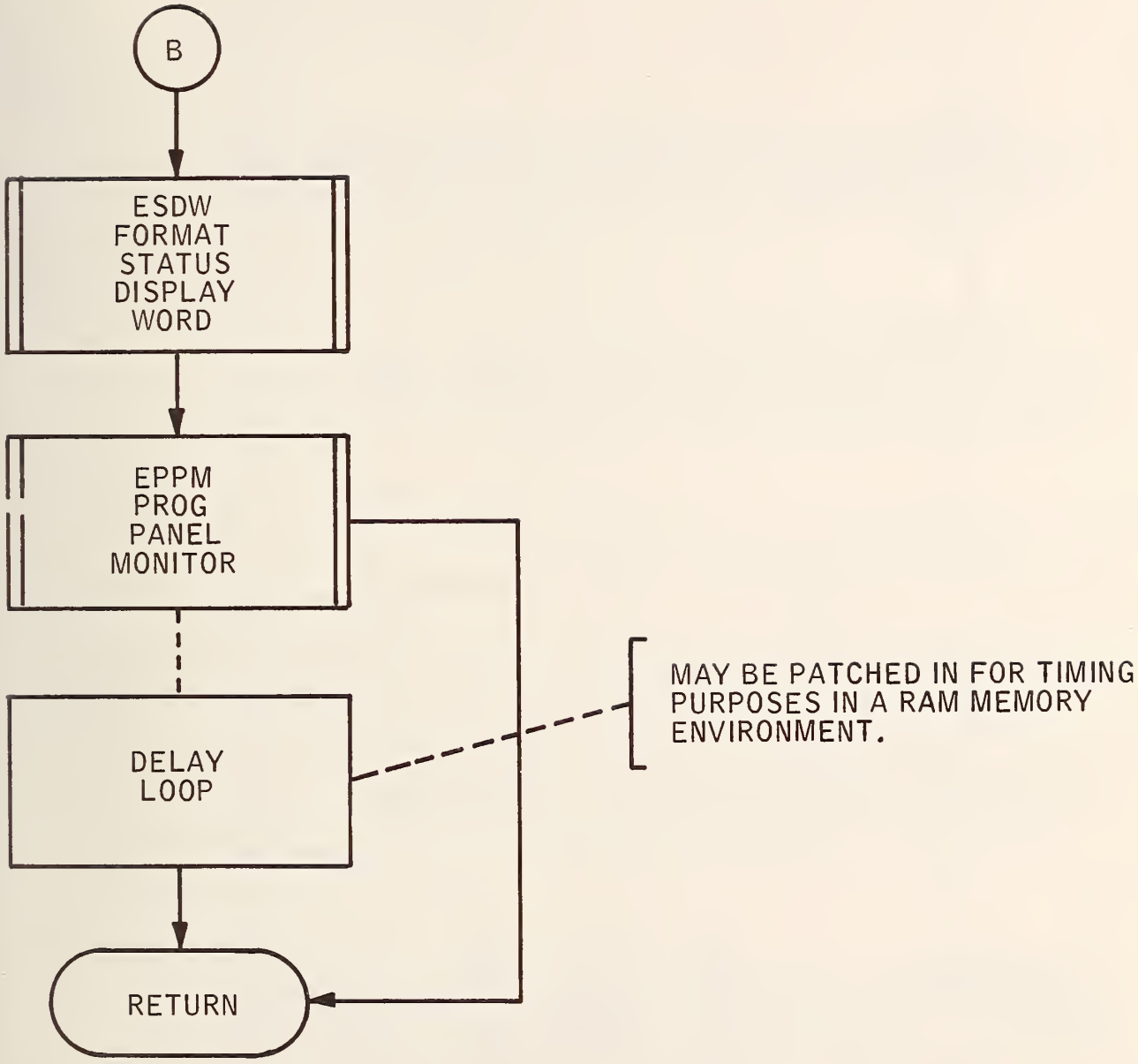


Figure B-7. Scheduling Routine (Executive CPC) (Concluded)

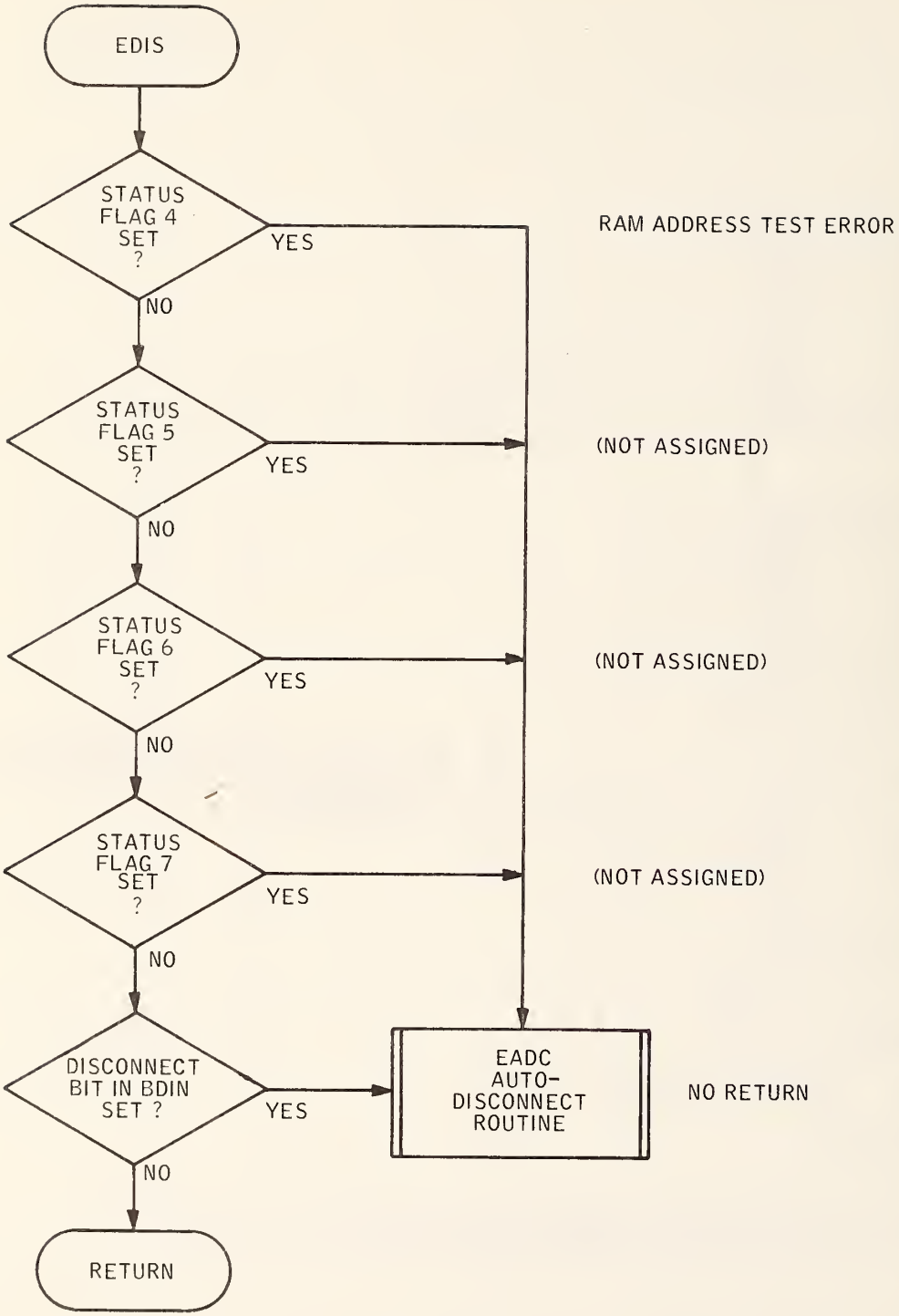


Figure B-8. Disconnect Monitor Routine (Executive CPC)

3.3.7 Command Monitoring Routine (ECOM)

3.3.7.1 Program Narrative

3.3.7.1.1 Purpose

This routine monitors the signal display command feedback from the LDU for agreement with the output commands. The watchdog timer is reset if the output commands agree with the feedback commands.

3.3.7.1.2 Software Interfaces

This routine is called only by the executive scheduling routine, ESCH, during each RTC interval. This routine calls the routine to reset the watchdog timer. See Table B-2 for a listing of the data fetched and the data stored by this routine.

3.3.7.1.3 Assumptions

Bits 15 and 14 in the command data words are "don't care" bits.

3.3.7.1.4 Solution Approach

Except for the "don't care" bits, the output command word is compared to the input command register status word for both main street and cross street signal display commands. If both commands are valid, the call is made to the routine to reset the watchdog timer. If either command is invalid, error flags are set and error data are recorded. An error flag, ECMF, triggers a subsequent output of the command word, which may remove the mismatch condition.

3.3.7.1.5 Program Operation

The current interval error flag is cleared. The main street and then the cross street command comparisons are made. Error data are recorded in the executive error buffer, EERR. The current interval error flag is set if either command is in error. If both commands are valid, the routine XWDT is called to output the reset control pulse to the watchdog timer.

3.3.7.1.6 Model

Not applicable.

3.3.7.1.7 Date: July 1975

3.3.7.1.8 Remarks

None.

3.3.7.2 Program Work Areas

The output command data are fetched from XOCW1 and XOCW2; the input command register data are fetched from BCR1 and BCR2. Hexadecimal constant, H3FFF, is used to mask off the "don't care" bits 15 and 14. The current interval error flag is ECMF. Error data are assigned to conditions 1 and 2 in the executive error buffer.

3.3.7.3 Program Flowchart

The program flowchart is presented in Figure B-9.

3.3.7.4 Program Listing

The program listing is presented in Appendix D.

3.3.7.5 Input/Output Data Formats

Not applicable.

3.3.7.6 Report Formats

Not applicable.

3.3.7.7 Test Data

To be determined.

3.3.7.8 Revisions

None.

3.3.8 Synchronization Routine (ESYN)

3.3.8.1 Program Narrative

3.3.8.1.1 Purpose

This routine synchronizes the software synchronization timer for the current timing plan with the loop master controller by means of the reset pulses on the reset select data lines.

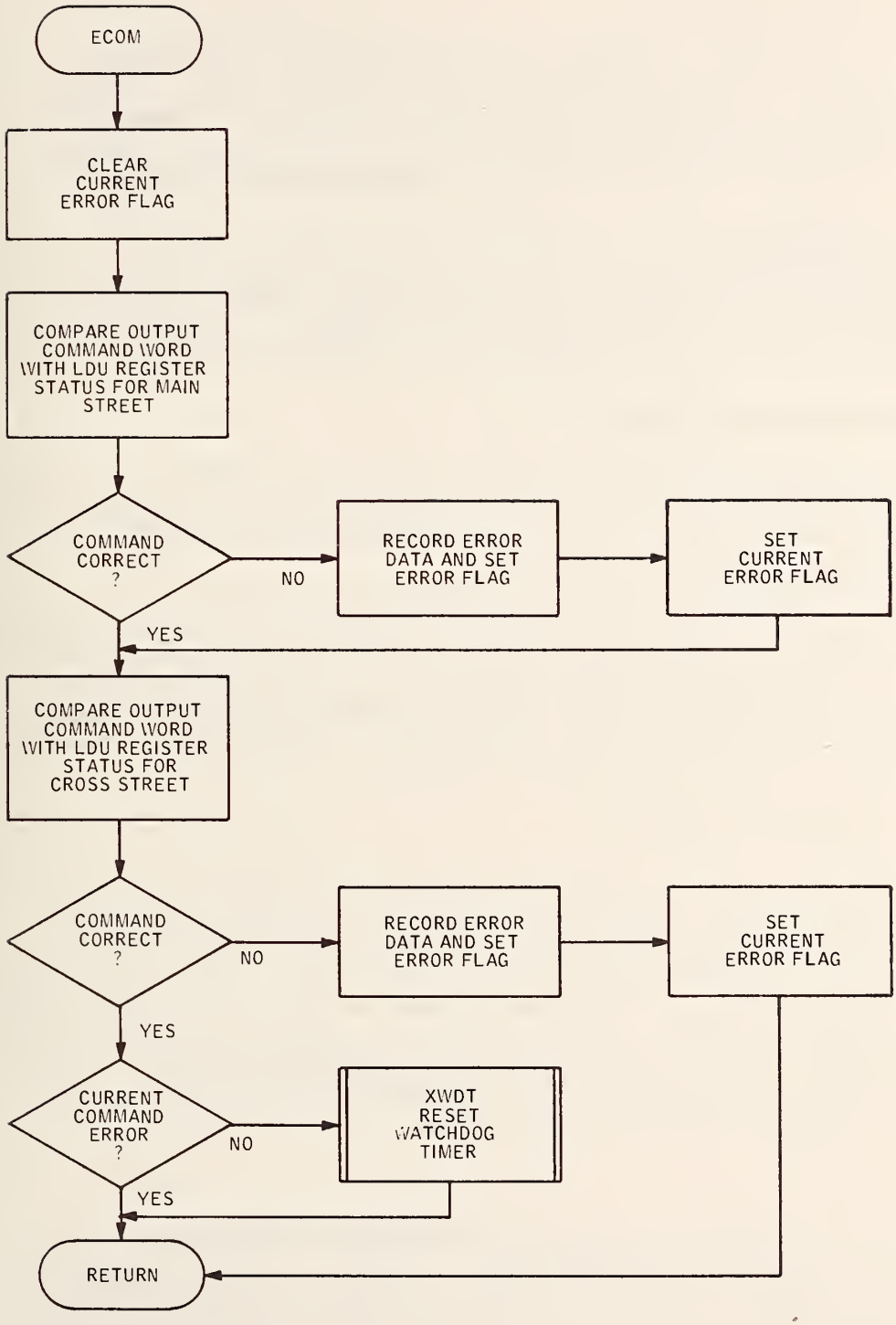


Figure B-9. Command Monitoring Routine (Executive CPC)

3.3.8.1.2 Software Interfaces

This routine is called only by the executive scheduling routine, ESCH, during each RTC interval. This routine calls an auxiliary routine ESYA to compute the base address of the synchronization data. See Table B-2 for a listing of the data stored and the data fetched by this routine.

3.3.8.1.3 Assumptions

To maintain synchronization, the cycle length set on the programmable panel thumbwheel switches must agree with the cycle length specified by the loop master controller.

3.3.8.1.4 Solution Approaches

The synchronization timer is incremented to the cycle length. At this time the start pulse is issued to the timing plan execution routine. The start pulse duration is 3 percent of the cycle length.

The timer is incremented once each RTC interval unless the reset line is high when the timer reaches that reset time. If the reset line is high, the timer is not incremented and essentially dwells until a pulse (low) appears on the reset line. Interrupter reset pulses are not distinguished from the master dial reset pulse.

3.3.8.1.5 Program Operation

The reset select datum is tested for having one or fewer resets high. If one reset is high, the synch dial timer is compared to the reset time. If the timer is at the reset time, the synch timer is not incremented. If the synch timer is not at the reset time or no reset is high, the synch timer is incremented. The synch timer is tested against the cycle length. At end of cycle, the start pulse is initiated. The synch timer is tested for exceeding 3 percent of the cycle length for ending the start pulse.

3.3.8.1.6 Model

The synch timer dwells at a reset point in the same manner as an electromechanical dial. However, as the timing plan will dwell only at the beginning of the cycle when waiting for a start pulse, the reset points may be anywhere within the cycle. An electromechanical dial is restricted as to where the resets may be placed.

3.3.8.1.7 Date: July 1975

3.3.8.1.8 Remarks

None

3.3.8.2 Program Work Areas

The reset select data are bits 1, 2, and 3 of ER123. The synchronization data are contained in blocks of six for each dial in buffer ESYD. The data are in the following sequence:

cycle length	reset 3
reset 1	end of start pulse (3 percent of cycle)
reset 2	synch dial timer.

3.3.8.3 Program Flowchart

The program flowchart is presented in Figure B-10.

3.3.8.4 Program Listing

The program listing is presented in Appendix D.

3.3.8.5 Input/Output Data Formats

Not applicable.

3.3.8.6 Report Formats

Not applicable.

3.3.8.7 Test Data

To be determined.

3.3.8.8 Revisions

None.

3.3.9 Interconnect Data Monitoring Routine (EMIC)

3.3.9.1 Program Narrative

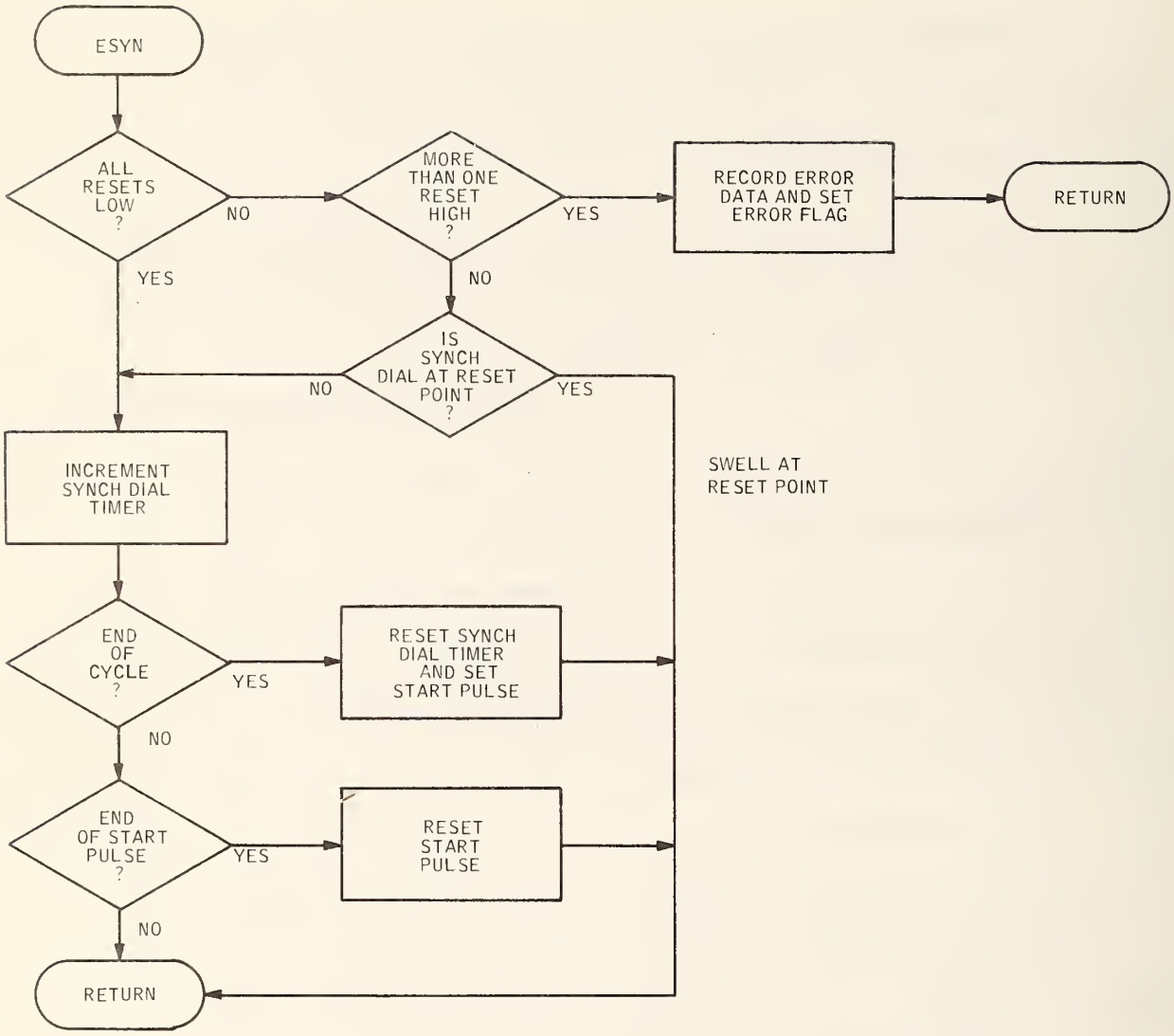


Figure B-10. Synchronization Routine (Executive CPC)

3.3.9.1.1 Purpose

This routine extracts the dial select and the reset select data from the discrete input data word and filters each set of data to provide short-term noise rejection. The source of the interconnect data is the loop master controller.

3.3.9.1.2 Software Interface

This routine is called only by the executive scheduling routine, ESCH, during each RTC interval. Table B-2 is a list of the data stored and the data fetched by this routine.

3.3.9.1.3 Assumptions

The dial select data D2, D3 are located in bits 4 and 5 and the reset select data R1, R2, and R3 are in bits 1, 2, and 3 of the discrete input data word, BDIN.

3.3.9.1.4 Solution Approach

Three past values of each data set are stored. If all three values are identical, the internal data words available to other functions are updated.

3.3.9.1.5 Program Operation

The dial select data bits are extracted and the three-element data buffer is updated to include the new datum. If all three values agree, the datum is updated to ED23 for reference by other functions.

The reset select data are then extracted, the buffer updated, and if all three values agree, the reset select datum, ER123, is updated.

3.3.9.1.6 Model

Not applicable.

3.3.9.1.7 Date: July 1975

3.3.9.1.8 Remarks

None.

3.3.9.2 Program Work Areas

New data are extracted from the discrete input data word, BDIN, stored in buffers starting at EDNO and ERNO and updated to the internal data words ED23 and ER123. Hexadecimal constants H000E and H0030 are used to extract the data bits.

3.3.9.3 Program Flowchart

The program flowchart is presented in Figure B-11.

3.3.9.4 Program Listing

The program listing is presented in Appendix D.

3.3.9.5 Input/Output Data Formats

Not applicable.

3.3.9.6 Not applicable.

3.3.9.7 Test Data

To be determined.

3.3.9.8 Revisions

None.

3.3.10 Timing Routine (ETIM)

3.3.10.1 Program Narrative

3.3.10.1.1 Purpose

This routine increments timers for the current cycle length, the dwell for synchronized start, and the detector monitoring intervals.

3.3.10.1.2 Software Interfaces

This routine is called only by the executive scheduling routine, ESCH, during each RTC interval. Table B-2 is a listing of the data fetched and the data stored by this routine.

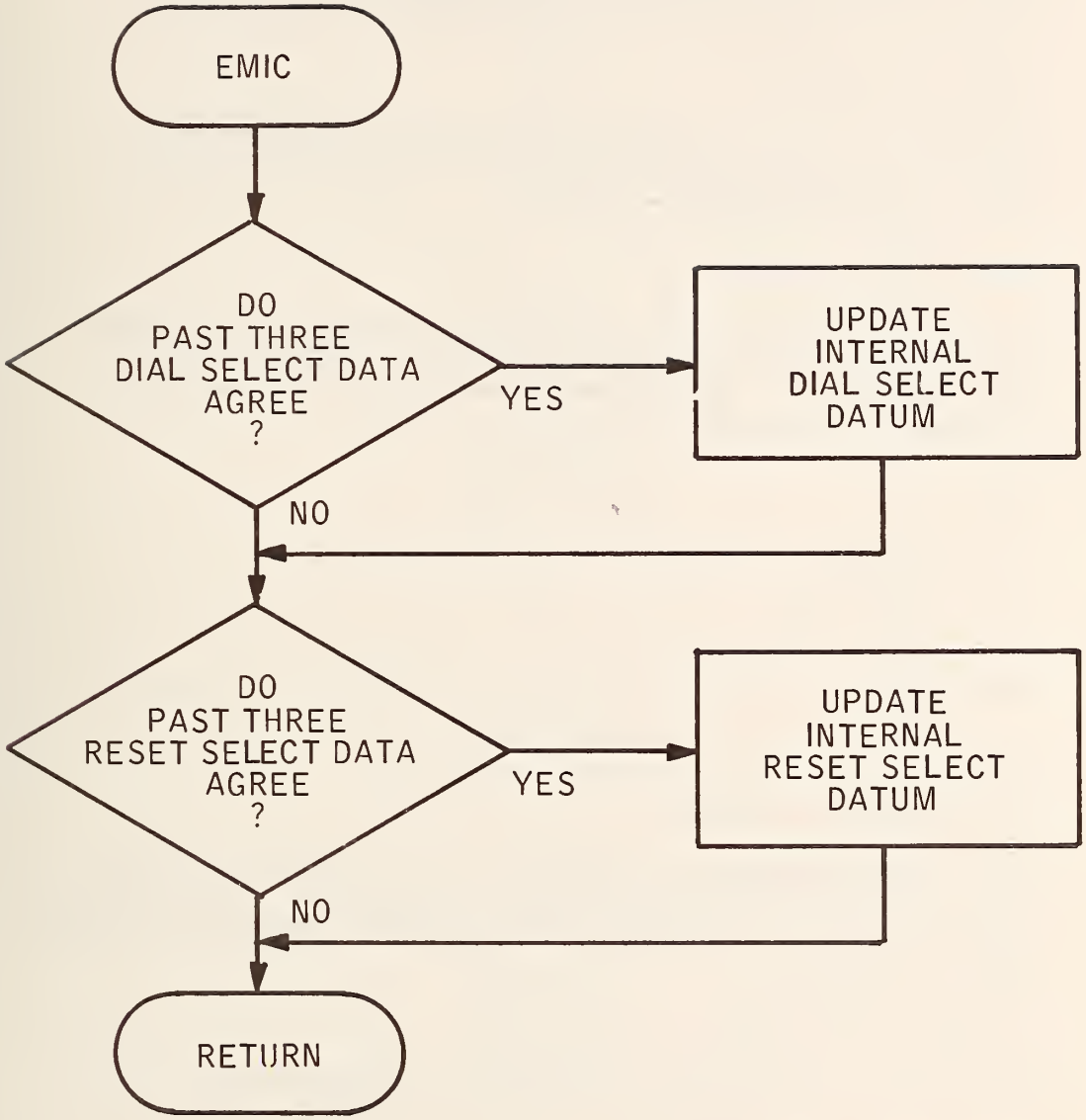


Figure B-11. Interconnect Data Monitor Routine (Executive CPC)

3.3.10.1.3 Assumptions

Other software functions are assumed to reset the timers to zero at the beginning of the appropriate time interval.

3.3.10.1.4 Solution Approach

If the timing plan execution is not in the dwell state for transfer of control, the cycle length and synch start dwell times are incremented by one.

3.3.10.1.5 Program Operation

If the WAIT/Control bit is set to the wait state, this routine does not increment the cycle length timer and the wait for synch start timer. The detector monitoring interval timer is always incremented however.

3.3.10.1.6 Model

Not applicable.

3.3.10.1.7 Date: July 1975

3.3.10.1.8 Remarks

None

3.3.10.2 Program Work Areas

The WAIT/CONTROL bit is bit 14 of the discrete input data word, BDIN. The cycle length timer is ETCY. The synchronization dwell timer is ETST. The detector monitoring interval timer is EDMIT.

3.3.10.3 Program Flowchart

The program flowchart is presented in Figure B-12.

3.3.10.4 Program Listing

The program listing is presented in Appendix D.

3.3.10.5 Input/Output Data Formats

Not applicable.

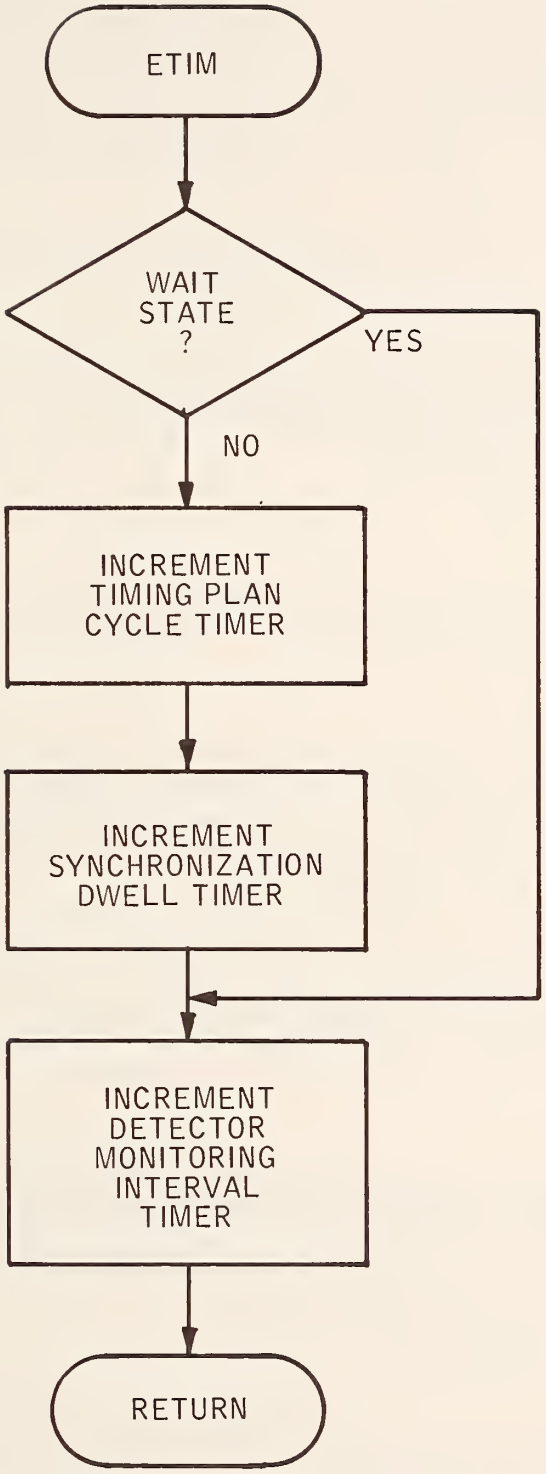


Figure B-12. Timing Routine (Executive CPC)

3.3.10.6 Report Formats

Not applicable.

3.3.10.7 Test Data

To be determined.

3.3.10.8 Revisions

None

3.3.11 End of Cycle Routine (EEOC)

3.3.11.1 Program Narrative

3.3.11.1.1 Purpose

This routine calls the programs to execute the various functions at the end of the timing plan cycle.

3.3.11.1.2 Software Interfaces

This routine is called only by the routine, TNXT, when the end of cycle condition is reached. This routine calls EPEOC, TNCY, and ETEOC to perform end of cycle functions for the Preempt CPC, the Timing Plan Execution CPC and the Executive CPC, respectively.

3.3.11.1.3 Assumptions

The timing plan is at the end of the cycle when this routine is called.

3.3.11.1.4 Solution Approach

The wait for synchronized start flag is first set but may be reset by the routines called by this routine. Synchronization dwell is allowed only at the beginning of the timing plan cycle and may be precluded by a preempt in progress, dial change, or programming panel update.

3.3.11.1.5 Program Operation

The wait for synch start flag is reset and then the end of cycle functions are executed.

Routine EPEOC performs the functions for the Preempt CPC. If a preempt modification was made during the past cycle, synchronization dwell is not allowed as a return to the equivalent synch status is guaranteed at the end of the next cycle. Preempt monitor index flags in data buffer, PMIF, are then modified to indicate the current cycle of the timing plan as opposed to the next cycle.

Routine TNCY performs the functions for the Timing Plan Execution CPC as documented elsewhere herein.

Finally ETEOC is called to perform the functions for the Executive CPC. The timing plan cycle timer is tested against the previous minimum and maximums and those data are updated as appropriate. The timer is then zeroed to start the next cycle.

3.3.11.1.6 Model

Not applicable.

3.3.11.1.7 Date: July 1975

3.3.11.1.8 Remarks

None

3.3.11.2 Program Work Areas

The preempt request flag, PFREQ, is tested for preempt modification in the past cycle. The wait for synch start flag, TFST, may be reset. Bit 15 in the three elements of PMIF are reset to indicate the current cycle. The cycle timer, ETCY, and maximum and minimum values are accessed and ETCY is reset to zero.

3.3.11.3 Program Flowchart

The program flowchart is presented in Figure B-13.

3.3.11.4 Program Listing

The program listing is presented in Appendix D.

3.3.11.5 Input/Output Data Formats

Not applicable.

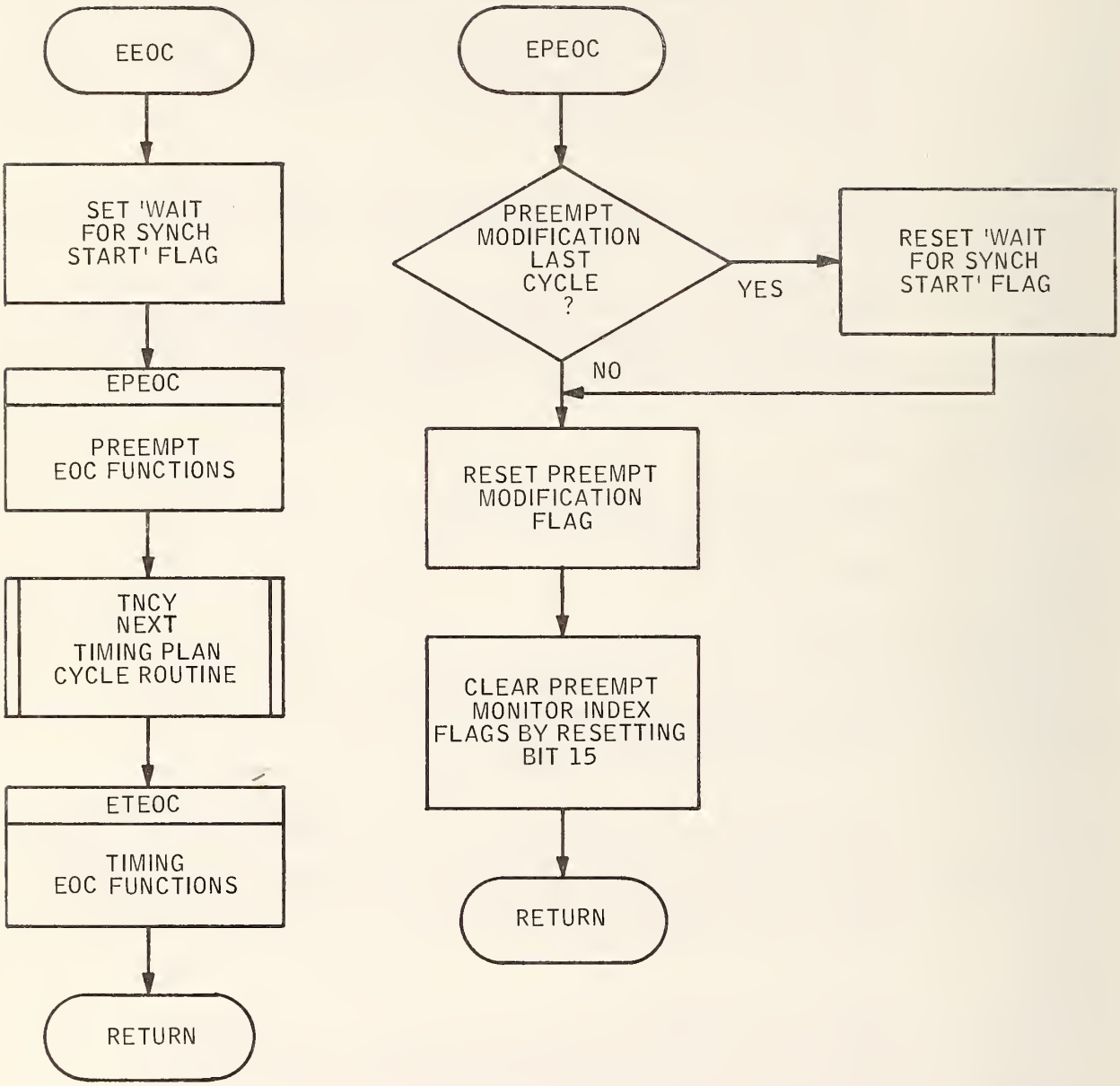


Figure B-13. End-of-Cycle Routine (Executive CPC)

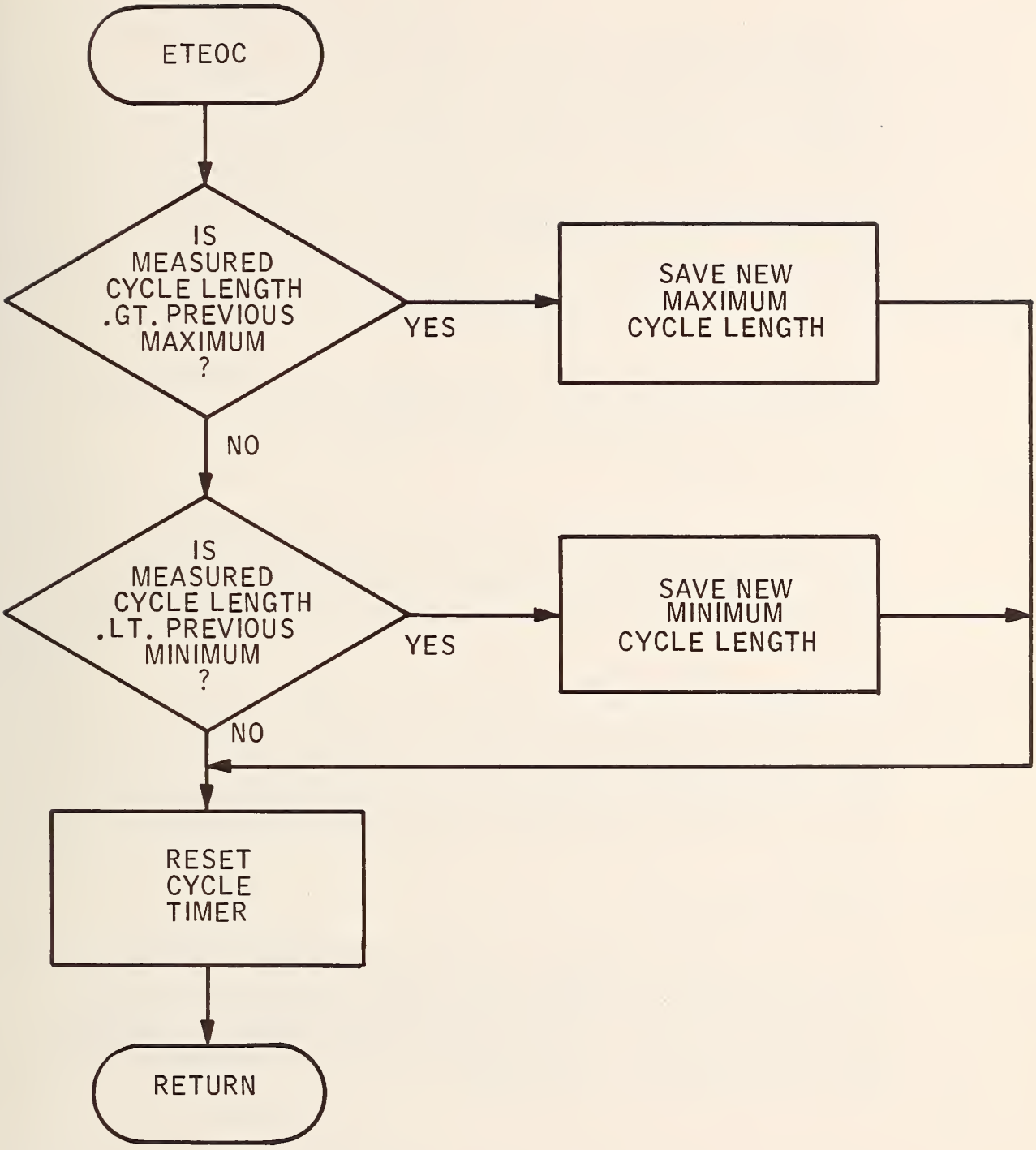


Figure B-13. End-of-Cycle Routine (Executive CPC) (Concluded)

3.3.11.6 Report Formats

Not applicable.

3.3.11.7 Test Data

To be determined.

3.3.11.8 Revisions

None

3.3.12 Status Display Routine (ESDW)

3.3.12.1 Program Narrative

3.3.12.1.1 Purpose

This routine constructs the status display word that drives the status lights on the SDU. The status display word output to the SDU is performed by other routines.

3.3.12.1.2 Software Interfaces

This routine is called by the executive scheduling routine, ESCH, during each RTC interval and by the autodisconnect routine, EADC, when a disconnect condition is determined. See Table B-2 for the data stored and the data fetched by this routine.

3.3.12.1.3 Assumptions

The reset select data are in bits 1, 2, and 3 of ER123. The dial select data are in bits 4 and 5 of ED23. All other bits in those data words are assumed to be zeroes. Any non-zero values in the other data tested will set the corresponding status bit.

3.3.12.1.4 Solution Approach

The status display contains 16 status lights that are driven by the 16-bit status display word. Bits 0-5 of the display are latching and require manual reset. Status conditions that may occur only once and be corrected via auto-restart are assigned to those latching bits. The remaining bits are updated by each output of the status display word.

3.3.12.1.5 Program Operation

The interconnect data used internally are extracted and stored with R1, R2, R3, D2, and D3 in bits 11 to 15, respectively. The start pulse state is stored in bit 10; the preempt in progress state is stored in bit 7. For the duration of the recent bus classification timer, VCBT, bit 6 will be on. Bit 5 is the aggregate of all software detected error conditions except detector undercounting, which is assigned to bit 4. Any error condition in each set will turn on the status light. Bit 2 displays the CMU conflict status. Bit 1 is the disconnect status external to the software. Bit 0 is the software disconnect status and is not processed by this routine.

3.3.12.1.6 Model

Not applicable.

3.3.12.1.7 Date: July 1975

3.3.12.1.8 Remarks

None

3.3.12.2 Program Work Areas

Status conditions are determined from the following data: ER123, ED23, EPST, PIP, VCBT, EERR, XERR, BERR, TERR, PERR, and BDIN. They are then stored temporarily in a register and are finally stored in the first word of the output buffer.

3.3.12.3 Program Flowchart

The program flowchart is presented in Figure B-14.

3.3.12.4 Program Listing

The program listing is presented in Appendix D.

3.3.12.5 Input/Output Data Formats

Not applicable.

3.3.12.6 Report Formats

Not applicable.

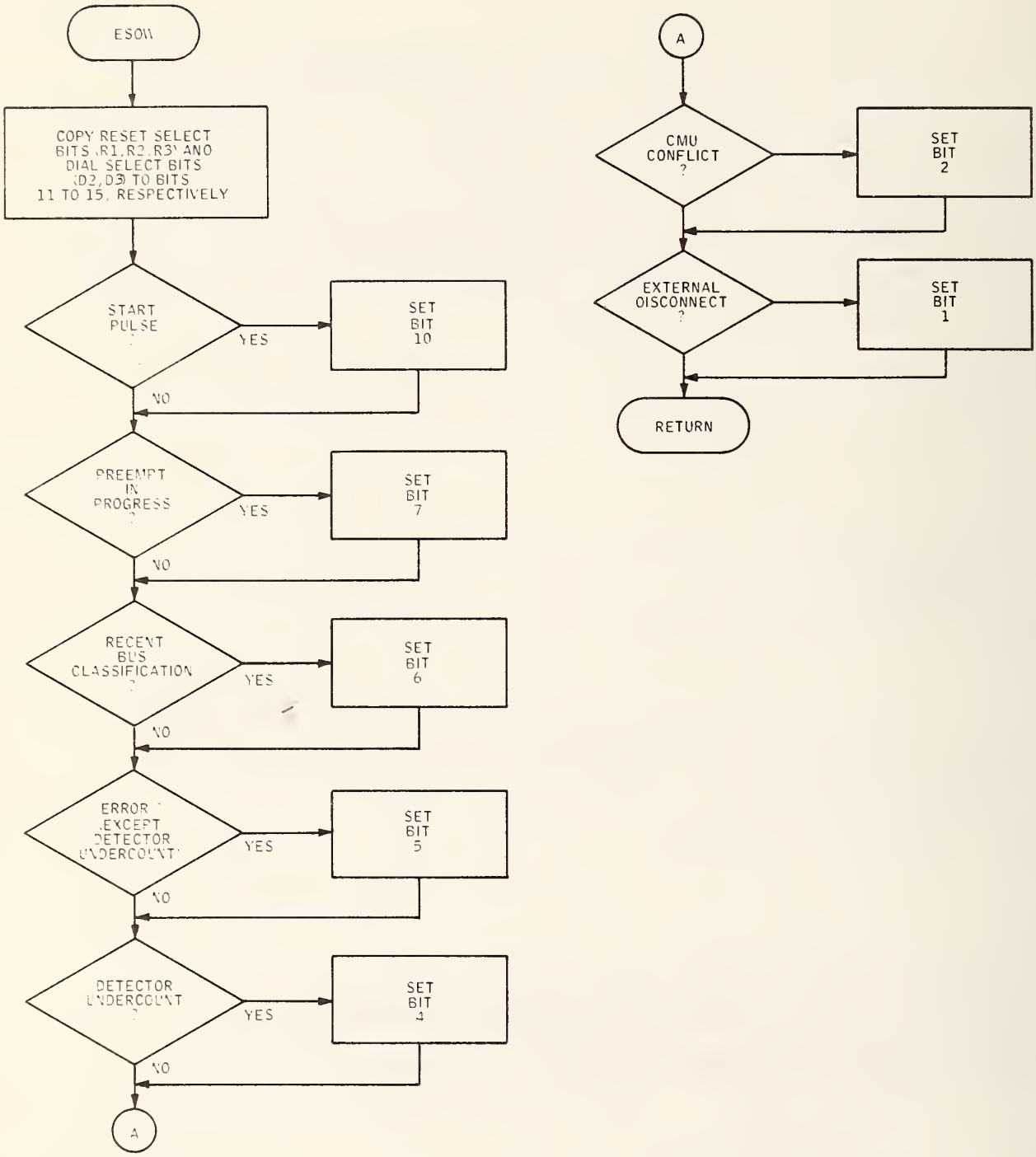


Figure B-14. Status Display Routine (Executive CPC)

3.3.12.7 Test Data

To be determined.

3.3.12.8 Revisions

None.

3.3.13.1 Detector Monitoring Routine (EDET)

3.3.13 Program Narrative

3.3.13.1.1 Purpose

This routine monitors the vehicle counts for each detector and records undercounting errors.

3.3.13.1.2 Software Interfaces

This routine is called only by the executive scheduling routine, ESCH, during each RTC interval. See Table B-2 for a listing of the data fetched and the data stored by this routine.

3.3.13.1.3 Assumptions

A timing routine is assumed to increment the timer for the detector monitoring interval. A maximum of 32 detectors is implied in a self-test feature, although only the actual number of detectors in the system are monitored for undercounting.

3.3.13.1.4 Solution Approach

At the end of each detector monitoring interval, the vehicle counters for each detector are tested and cleared to zero. Any vehicle counter not exceeding the minimum count results in error data recorded in the error buffer.

3.3.13.1.5 Program Operation

The interval timer is tested for end of interval. At the end of the interval, the timer is cleared to zero and the vehicle counters for all detectors in the system are tested and undercount failures are recorded.

3.3.13.1.6 Model

Not applicable.

3.3.13.1.7 Date: July 1975

3.3.13.1.8 Remarks

None.

3.3.13.2 Program Work Areas

The detector monitoring interval timer, EDMIT, is tested and cleared to zero. Vehicle counters are contained in array, VCVC. A temporary storage location, EDETT, is used during the error recording process. Error data are stored in both the executive error buffer, EERR, and the detector error buffer, EDERR.

Local data values define the duration of the detector monitoring interval and the vehicle count threshold.

3.3.13.3 Program Flowchart

The program flowchart is presented in Figure B-15.

3.3.13.4 Program Listing

The program listing is presented in Appendix D.

3.3.13.5 Input/Output Data Formats

Not applicable.

3.3.13.6 Report Formats

Not applicable.

3.3.13.7 Test Data

To be determined.

3.3.13.8 Revisions

None.

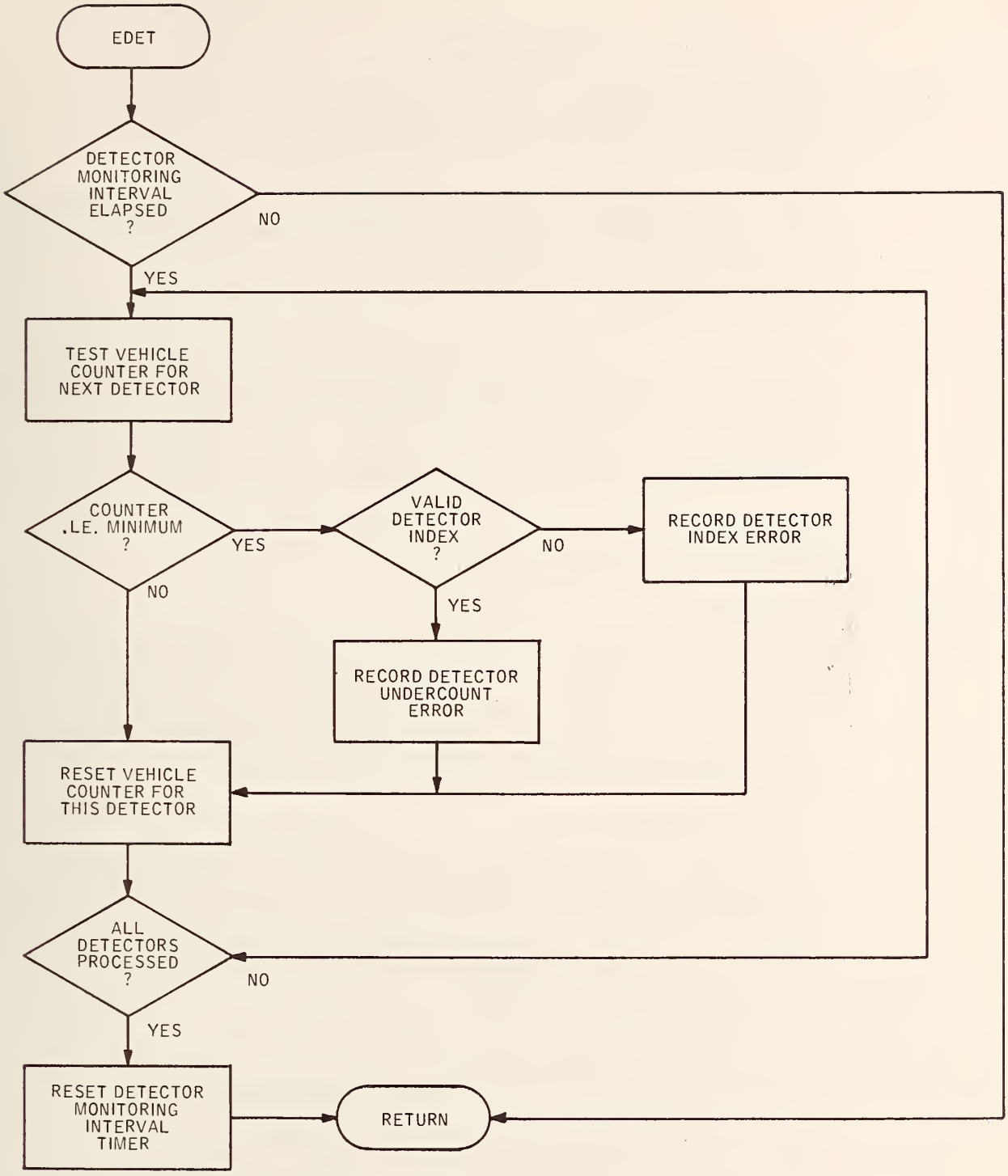


Figure B-15. Detector Monitoring Routine (Executive CPC)

3.3.14 Autodisconnect Routine (EADC)

3.3.14.1 Program Narrative

3.3.14.1.1 Purpose

This routine performs the functions preliminary to the software autodisconnect response to critical conditions that preclude further system operation.

3.3.14.1.2 Software Interfaces

This routine is called by those routines that diagnose critical conditions precluding further system operation. The EADC routine calls ESDW to set up the status display word for the conditions at the autodisconnect. The routine ends in a terminal wait loop and never returns to the calling program.

See Table B-2 for a listing of the data fetched and the data stored by this routine.

3.3.14.1.3 Assumptions

This routine assumes that the external disconnect bit in the discrete input word is set for autodisconnect conditions external to the software diagnostic tests. If the external disconnect bit is not set, this routine assumes that a software diagnosed critical condition has been sensed.

3.3.14.1.4 Solution Approach

The computational environment at the time of the disconnect is saved. The status display word is constructed and output to the SDU, and the program enters a terminal wait loop. The terminal wait loop maintains the CPU in a running mode as opposed to halted, while preserving the RAM data in the status at the autodisconnect event.

3.3.14.1.5 Program Operation

The accumulators are saved, the real time clock is stopped, and the interrupt request is assured to be reset. The interrupt enable flags, stack and select flag are saved. The status display word is then constructed except for the internal disconnect bit. The source of the autodisconnect is determined by the state of the external disconnect bit in the status display word as extracted from the discrete input word. For an

internal disconnect, the output control pulse to the transfer control panel is issued. The status display word is then output to the SDU and the program enters the terminal wait loop.

3.3.14.1.6 Model

Not applicable.

3.3.14.1.7 Date: July 1975

3.3.14.1.8 Remarks

None.

3.3.14.2 Program Work Areas

The computational environment of register contents, RALU status flags, CPU control flags and stack are saved in the disconnect save area, EDSAV. The status display word is constructed by ESDW in the same manner as during the RTC sequence. The state of the internal disconnect bit is determined by this routine however.

3.3.14.3 Program Flowchart

The program flowchart is presented in Figure B-16.

3.3.14.4 Program Listing

The program listing is presented in Appendix D.

3.3.14.5 Output Data Formats

The format of the status display word is defined in Table B-8. The status display word output is performed by this routine. Also, the output control pulses issued by this routine are completely defined by the peripheral device address and function code and do not rely on the contents of Accumulator 0.

3.3.14.6 Report Formats

Not applicable.

3.3.14.7 Test Data

To be determined.

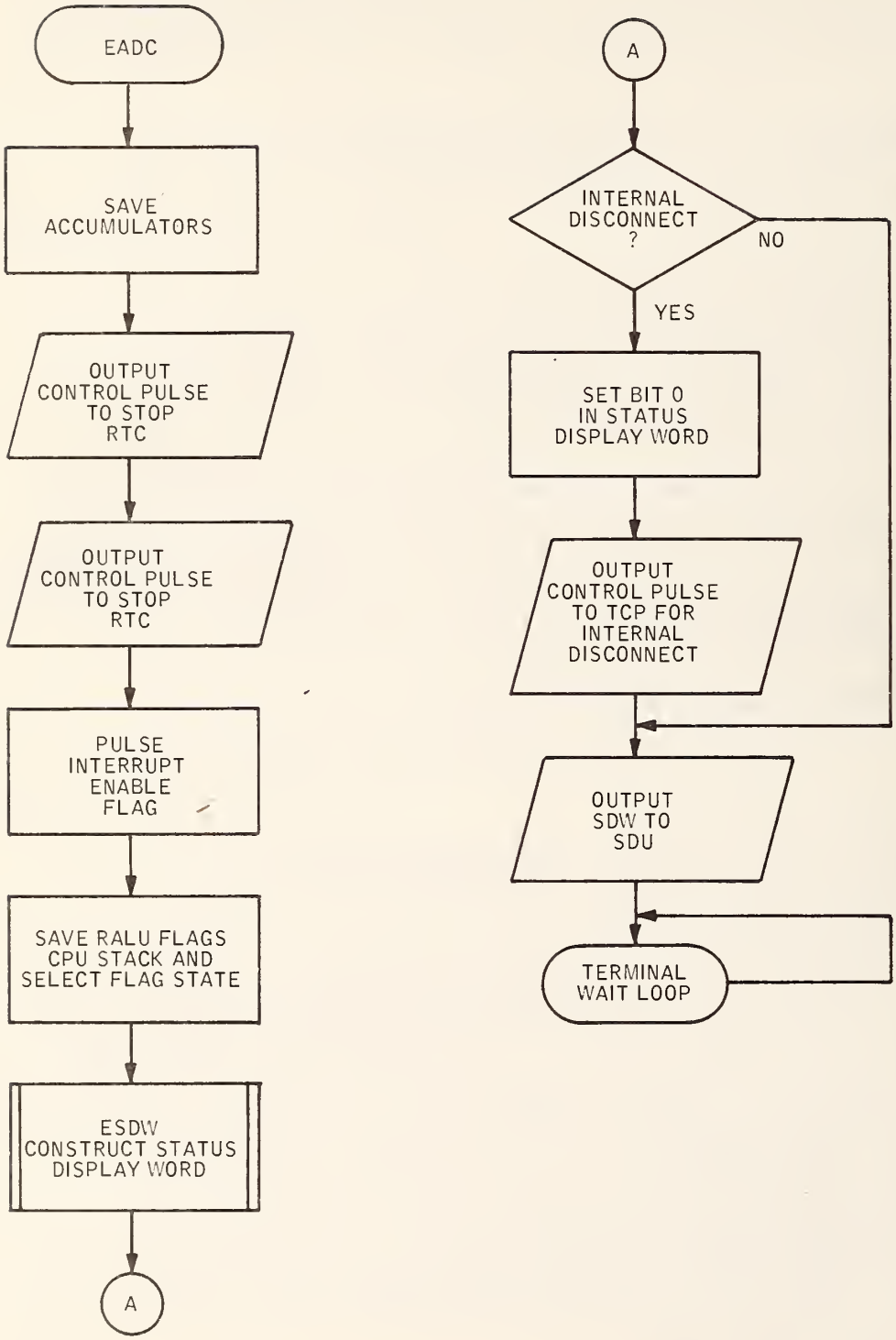


Figure B-16. Autodisconnect Routine (Executive CPC)

3.3.14.8 Revisions

None.

3.3.15 Program Narrative

3.3.15.1.1 Purpose

This routine controls the initialization of all data contingent upon the programming panel settings.

3.3.15.1.2 Software Interfaces

This routine is called only by ERSI during the restart initialization sequence. This routine calls the panel input routine BPAN; the routines to process the set of data for each timing plan EPP1, EPP2, and EPP3; the data base update routine EPUP; the routines to turn panel status lights on and off XLON1, XLOF1, XLON2, XLOF2; the routine to compute the sum of the fixed intervals, EPFI; and, in case of fatal error, the error recording routine ERE0; the reset watchdog timer routine, XWDT; and the autodisconnect routine, EADC. Table B-2 is a list of the data fetched and the data stored by this routine.

3.3.15.1.3 Assumptions

Not applicable.

3.3.15.1.4 Solution Approach

This routine controls the entire initialization sequence of the programming panel data. The timing plan tables are checked to assure that the adjustable interval durations are initially zero as a test on data base consistency. The sum of the fixed intervals are then computed and stored for later use by EPP1, EPP2, and EPP3. The data is then processed for each of the three timing plans. An adjustable interval duration error or a data validity error during initialization is fatal and results in autodisconnect. The watchdog timer reset pulse is issued to assure that the timer is started for additional autodisconnect mechanization.

3.3.15.1.5 Program Operation

The input routine is called. The local error flags and error status lights are cleared. The sum of fixed intervals is computed and stored for each plan. The adjustable interval

durations are tested for zero. The timing plan data sets are then processed. Error flags are then tested and the panel status lights are turned on for any errors. For no data validity error in any data set, the data base is initialized and the pending input request light is turned off.

3.3.15.1.6 Model

Not applicable.

3.3.15.1.7 Date: July 1975

3.3.15.1.8 Remarks

None.

3.3.15.2 Program Work Areas

The error flags EP1E, EP2E, EP3E, and EPDVE are cleared. The sums of fixed intervals are stored in EPFT, and relative locations of adjustable intervals, TAI1 and TAI2, are used to access the adjustable interval durations in the timing plan interval duration table. Error buffers EPEF1, EPEF2, and EPEF3 contain any error data.

3.3.15.3 Program Flowchart

The program flowchart is presented in Figure B-17.

3.3.15.4 Program Listing

The program listing is presented in Appendix D.

3.3.15.5 Input/Output Data Formats

Not applicable.

3.3.15.6 Report Formats

Not applicable.

3.3.15.7 Test Data

To be determined.

3.3.15.8 Revisions

None.

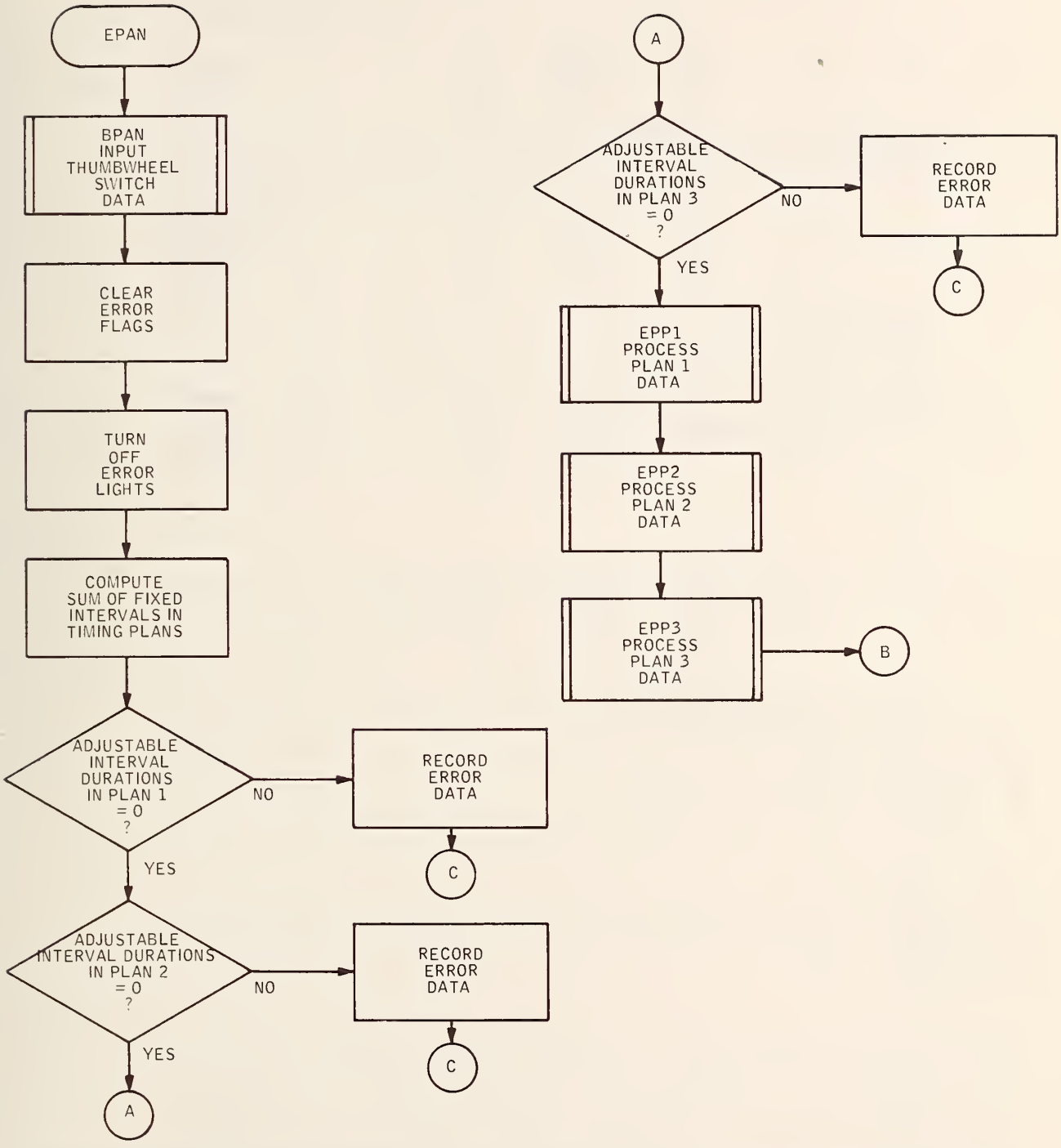


Figure B-17. Programming Panel Initialization Routine (Executive CPC)

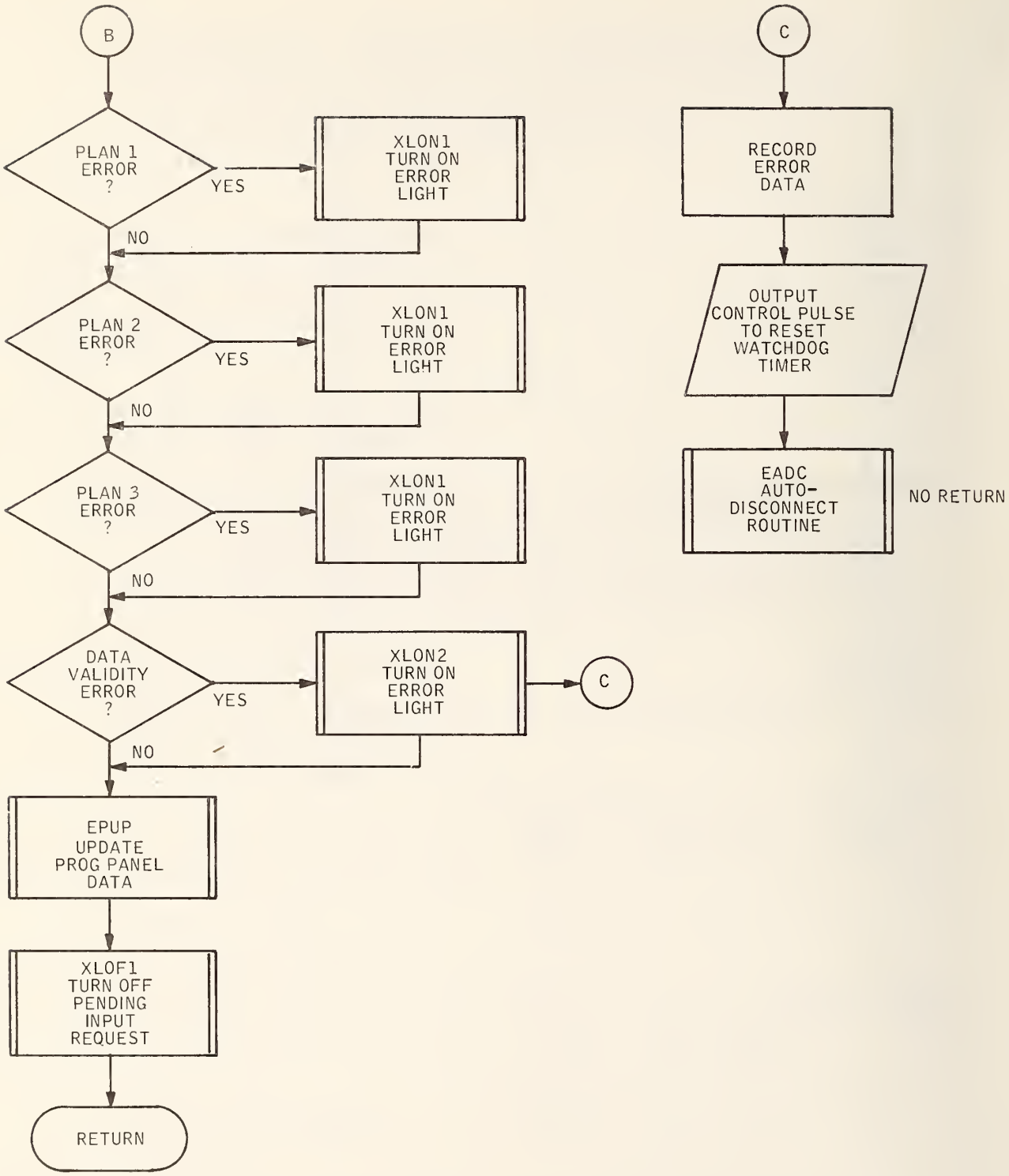


Figure B-17. Programming Panel Initialization Routine (Executive CPC) (Concluded)

3.3.16 Programming Panel Real Time Monitor Routine (EPPM)

3.3.16.1 Program Narrative

3.3.16.1.1 Purpose

This routine monitors the input request from the panel and controls the sequence of panel data input, processing of data, and indication of pending update of data base parameters determined by the panel data.

3.3.16.1.2 Software Interfaces

This routine is called by the executive scheduling routine, ESCH, during each real time clock interval. This routine calls the panel data input routine, BPAN; the routine to turn panel status lights on and off XLON1, XLOF1, XLON2, XLOF2; and the timing plan data processing routines EPP1, EPP2, and EPP3. Table B-2 is a list of the data fetched and the data stored by this routine.

3.3.16.1.3 Assumptions

The input request push-button state is assumed to be bit 10 in the programming panel discrete input word.

3.3.16.1.4 Solution Approach

The input request push-button state is monitored for a state change from 0 to 1. This initiates the following sequence over the next four real time clock intervals: input thumbwheel switch data, process plan 1 data, process plan 2 data, and process plan 3 data. The panel update flag is set to trigger data base update at the next end-of-timing-plan-cycle event.

The computational load is spread over the four real time clock intervals to reduce the maximum overhead for this nonperiodic computation requirement in the real time sequence.

3.3.16.1.5 Program Operation

The input request is monitored for a state change. The change from 0 to 1 triggers the processing sequence discussed in 3.3.16.1.4. For the 0 state, the data base update flag is assured to be reset to allow the operator to cancel an input request, for example, in the event that errors are diagnosed and error lights are lit following the input request. The input request button is push on/push off from the operator's standpoint.

When the processing sequence is initiated, the error flags are cleared and the thumbwheel data is input. The computation flag is set for plan 1 processing and the program exits.

In the next interval, the plan 1 error light is turned off, the data processing error flag is tested, the error light is turned on in the event of an error, the computation flag is set for plan 2 processing and the routine exits.

Plan 2 is processed in the third interval and plan 3 in the fourth interval. At the end of the sequence, the update flag is set and the computation flag is reset to the null state. A data validity error precludes update of the data base.

3.3.16.1.6 Model

Not applicable.

3.3.16.1.7 Date: July 1975

3.3.16.1.8 Remarks

None.

3.3.16.2 Program Work Areas

The error flags EP1E, EP2E, EP3E, and EPDVE are cleared. The data base update flag EPUF is set and cleared. The computation sequence flag is incremented at each phase of the computation sequence and reset at the end of the sequence.

3.3.16.3 Program Flowchart

The program flowchart is presented in Figure B-18.

3.3.16.4 Program Listing

The program listing is presented in Appendix D.

3.3.16.5 Input/Output Data Formats

Not applicable.

3.3.16.6 Report Formats

Not applicable.

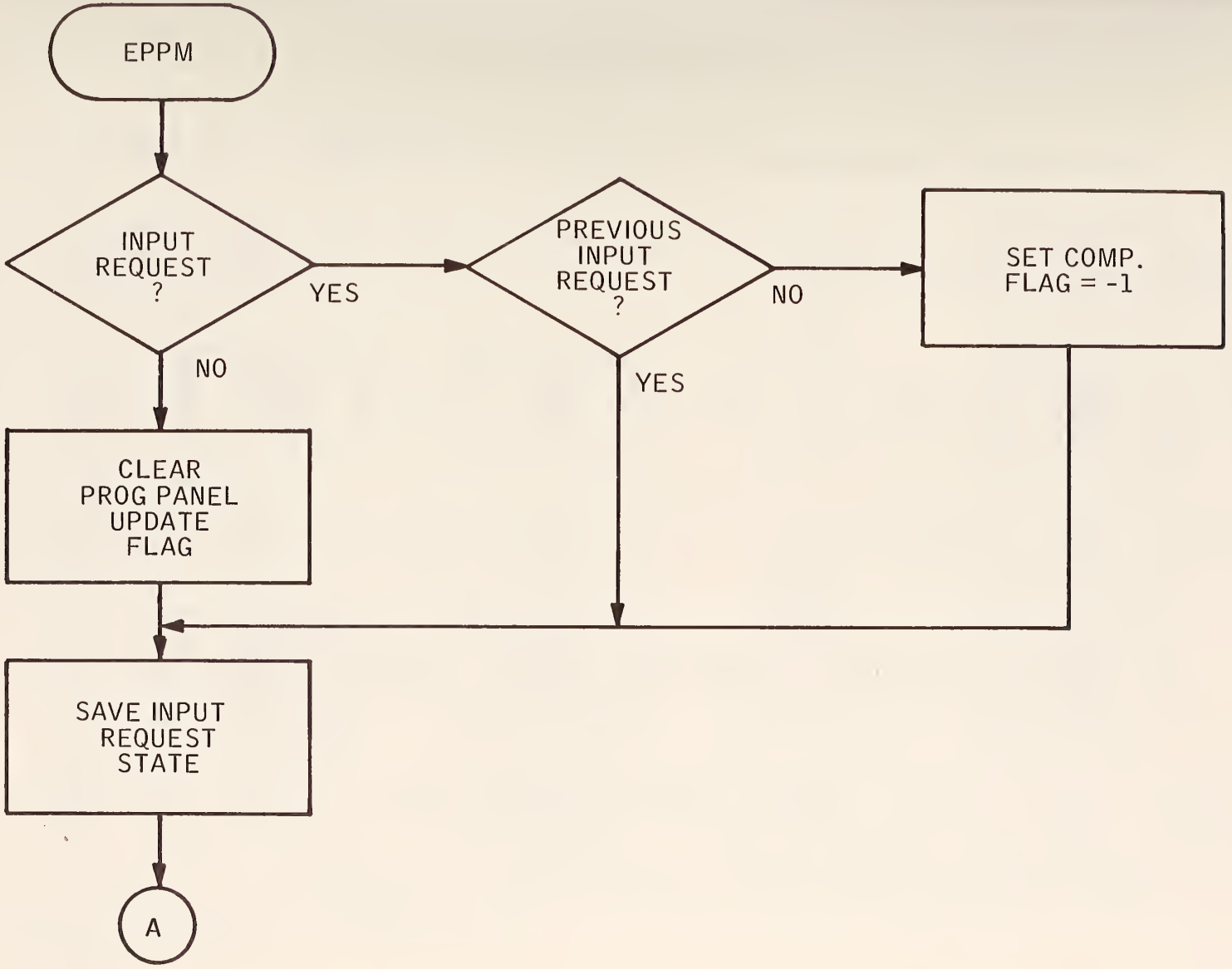


Figure B-18. Programming Panel Real Time Monitor Routine (Executive CPC)

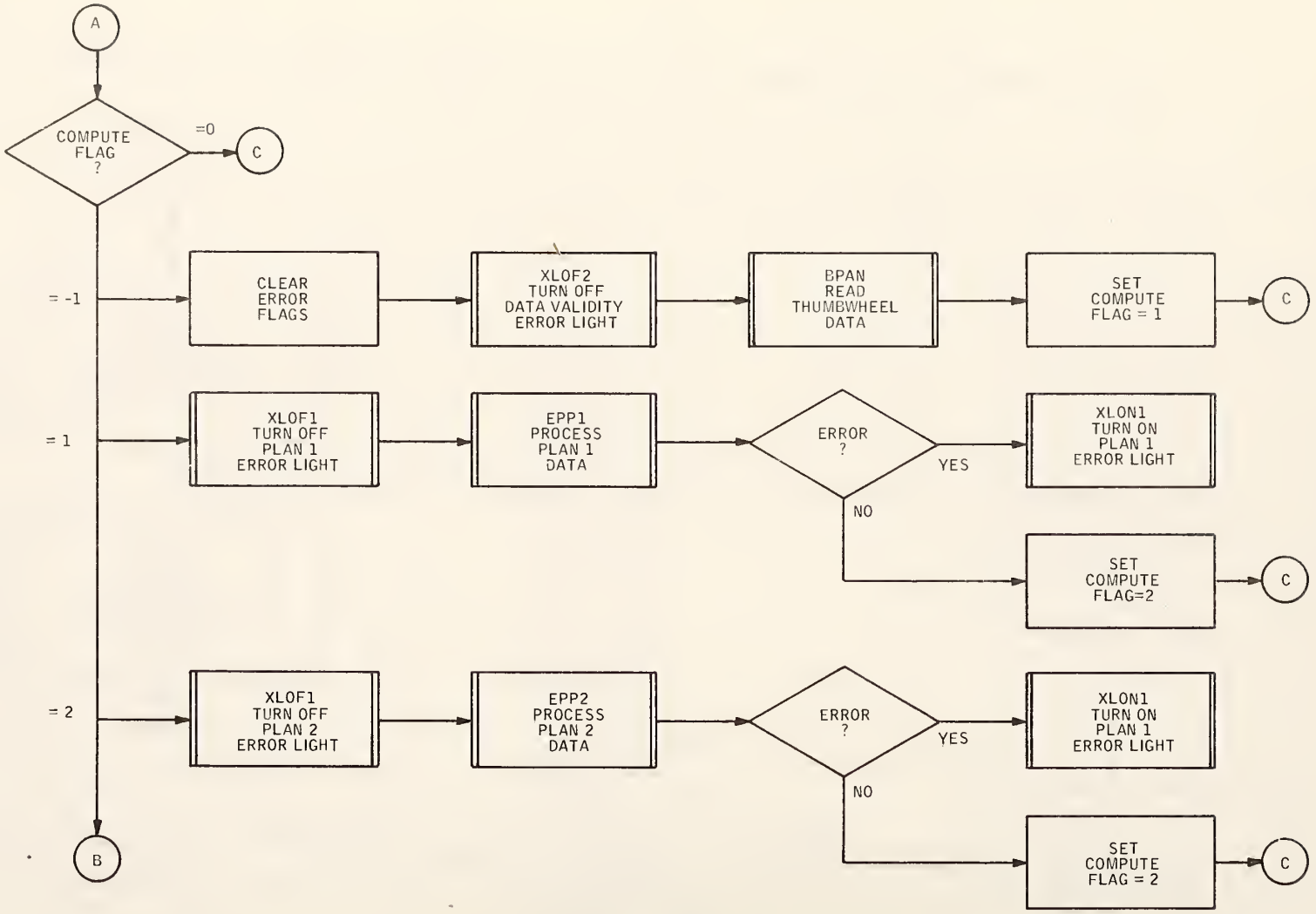


Figure B-18. Programming Panel Real Time Monitor Routine (Executive CPC) (Continued)

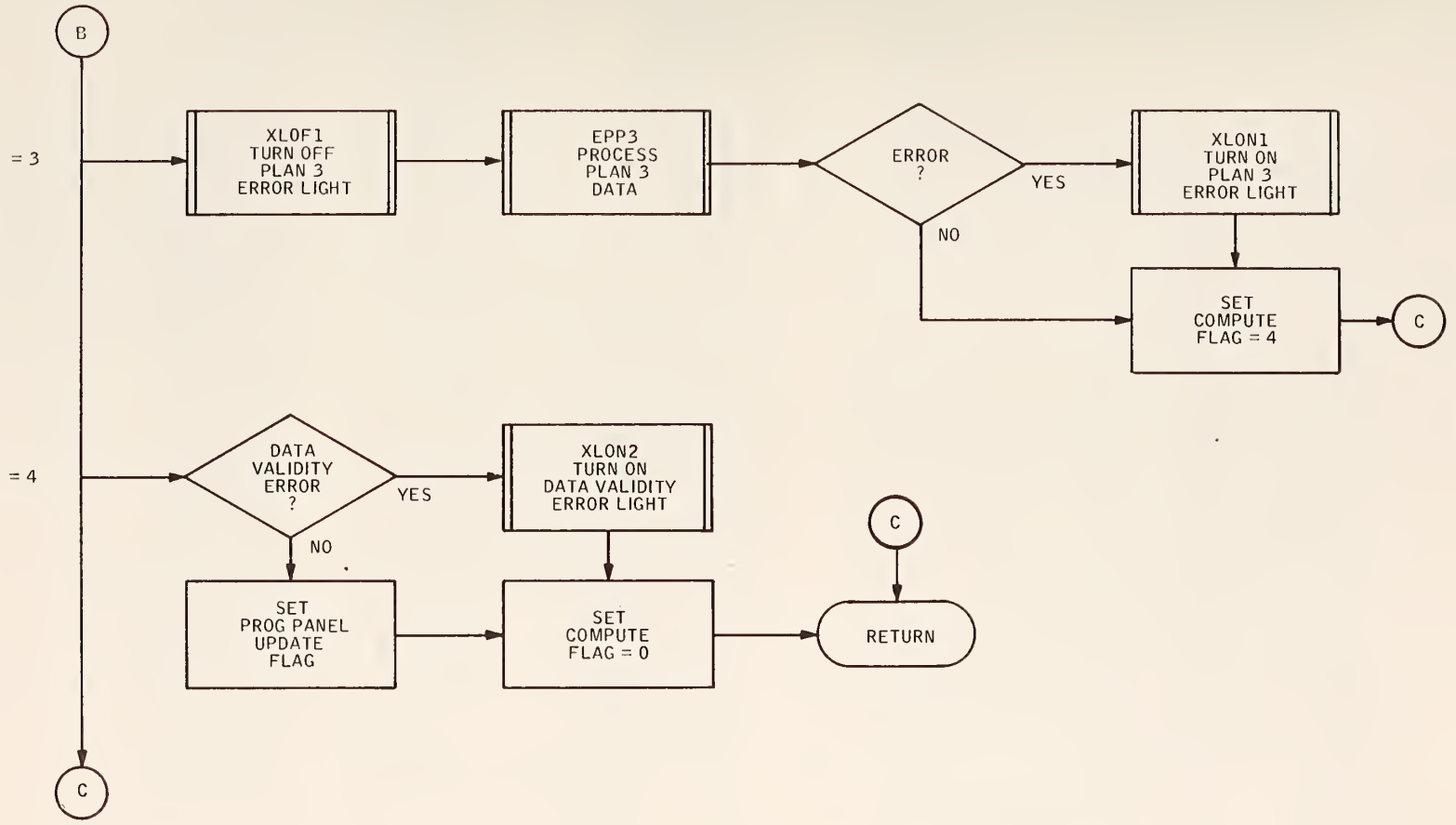


Figure B-18. Programming Panel Real Time Monitor Routine (Executive CPC) (Concluded)

3.3.16.7 Test Data

To be determined.

3.3.16.8 Revisions

None.

3.3.17 Process Programming Panel Plan 1 Routine (EPP1)

3.3.17.1 Program Narrative

3.3.17.1.1 Purpose

This routine processes the raw data input from the plan 1 thumbwheel switches. Data values are converted from binary code to integers, checked for data validity (digits 0-9), data reasonableness, and data consistency. Processed data values are stored in a data base buffer for subsequent transfer to the active portions of the data base.

3.3.17.1.2 Software Interfaces

This routine is called by EPAN during restart initialization and upon demand by EPPM during real time operation. This routine calls the data conversion routines EPBI2 and EPBI3; the error recording routines ERE0-ERE12; and multiplication routines EM10 and EM24. Table B-2 is a list of the data fetched and the data stored by this routine.

3.3.17.1.3 Assumptions

Maximum synchronization dwell time is 24 percent of the cycle length. The start pulse at MSG is 3 percent of the cycle length. Real time clock interval is 0.1 seconds. Thumbwheel switch data units except for MSG split percent are in seconds.

3.3.17.1.4 Solution Approach

The program processes the data in the following order: cycle length, offset 1, offset 2, offset 3, MSG split percent, and INT1. Each digit is tested for validity before conversion, and each converted value is tested for reasonableness and consistency, as appropriate.

3.3.17.1.5 Program Operation

The cycle is converted and tested against the maximum cycle length parameters and the sum of the fixed intervals in the timing plan. Out-of-range values are limited. The adjustable time (excess of cycle length over the sum of the fixed intervals) is computed and stored temporarily. The maximum dwell time for synchronization and the start pulse duration are computed. The offsets are converted and limited to a maximum equal to the cycle length. The reset time is computed as the cycle length complement of the offset time. The MSG split percent is then converted and the adjustable interval durations are computed. The INT1 parameter is converted, tested against a maximum, and converted to RTC units.

3.3.17.1.6 Model

Not applicable.

3.3.17.1.7 Date: July 1975

3.3.17.1.8 Remarks

None.

3.3.17.2 Program Work Areas

Raw input data is fetched from the input buffer, BPD1; converted data are stored in the buffer, EPCD1; computed data base parameters are stored in the buffer, EPD1; and error data are stored in EPEF1. Error flags EP1E and EPDVE are set as appropriate.

3.3.17.3 Program Flowchart

The program flowchart is presented in Figure B-19.

3.3.17.4 Program Listing

The program listing is presented in Appendix D.

3.3.17.5 Input/Output Data Formats

Not applicable.

3.3.17.6 Report Formats

Not applicable.

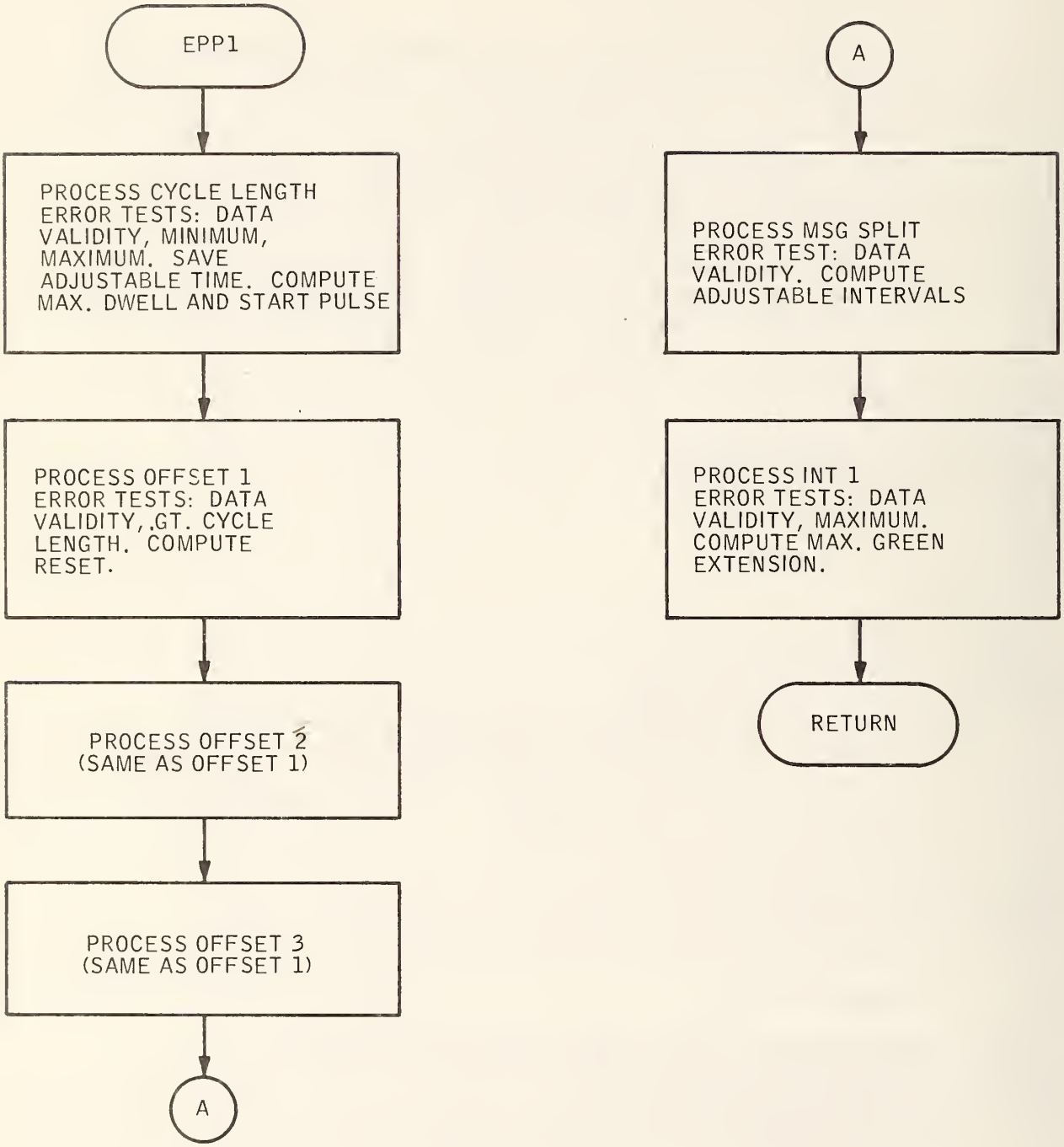


Figure B-19. Process Programming Panel Plan 1 Data Routine (Executive CPC)

3.3.17.7 Test Data

To be determined.

3.3.17.8 Revisions

None.

3.3.18 Process Programming Panel Plan 2 Routine (EPP2)

This routine is identical to EPP1 except for the data buffers and error flag locations. This routine references input buffer BPD2, converted data buffer EPCD2, and processed data buffer EPD2. Error flags EP2E and EPDVE are set as appropriate. See Figure B-20.

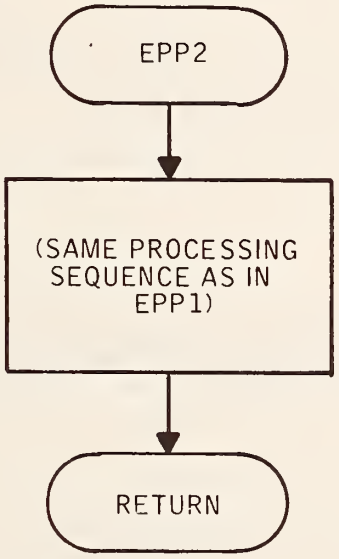


Figure B-20. Process Programming Panel Plan 1 Data Routine (Executive CPC)

3.3.19 Process Programming Panel Plan 3 Routine (EPPE)

This routine is identical to EPP1 except for the data buffers and error flag locations. This routine references input buffer BPD3, converted data buffer EPCD3, and processed data buffer EPD3. Error flags EP3E and EPDVE are set as appropriate. See Figure B-21.

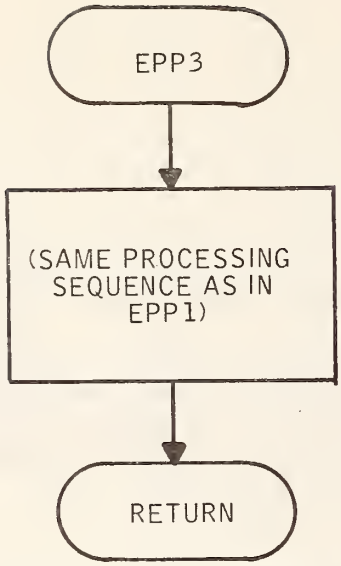


Figure B-21. Process Programming Panel Plan 3 Data Routine (Executive CPC)

3.3.20 Update Programming Panel Data Routine (EPUP)

3.3.20.1 Program Narrative

3.3.20.1.1 Purpose

This routine transfers data from the buffers for the programming panel routines to the locations in the active data base.

3.3.20.1.2 Software Interfaces

This routine is called by EPAN during initialization and by TNCY upon panel update during real time operation. Table B-2 is a list of the data stored and the data fetched by this routine.

3.3.20.1.3 Assumptions

This routine uses assumed sequences of data in the programming panel buffers and active data arrays to transfer data. End of timing plan cycle is assumed in the processing of synchronization parameters. The synchronization data blocks are assumed to be six locations in length.

3.3.20.1.4 Solution Approach

This routine uses the assumed data sequences and address pointers to transfer data. The synchronization dial timer

is set to the new cycle length such that end of cycle will be recognized by the synchronization routine that is executed subsequent to this routine at end of timing plan cycle.

3.3.20.1.5 Program Operation

The adjustable intervals are stored in the master timing plan tables. Then the maximum synchronization dwell parameters and the maximum preempt extensions are updated. Finally, the synchronization parameters for each timing plan are updated and the synchronization dial timers are set to the new cycle length.

3.3.20.1.6 Model

Not applicable.

3.3.20.1.7 Date: July 1975

3.3.20.1.8 Remarks

None.

3.3.20.2 Program Work Areas

The new data values are fetched from buffers EPCD1, EPCD2, and EPCD3. Adjustable interval durations are stored in the timing plan tables TPM1, TPM2, and TPM3; maximum synchronization dwell parameters are stored in TDWL; maximum preempt extension parameters are stored in PEMAX; and synchronization parameters are stored in ESYD in blocks of six.

3.3.20.3 Program Flowchart

The program flowchart is presented in Figure B-22.

3.3.20.4 Program Listing

The program listing is presented in Appendix D.

3.3.20.5 Input/Output Data Formats

Not applicable.

3.3.20.6 Report Formats

Not applicable.

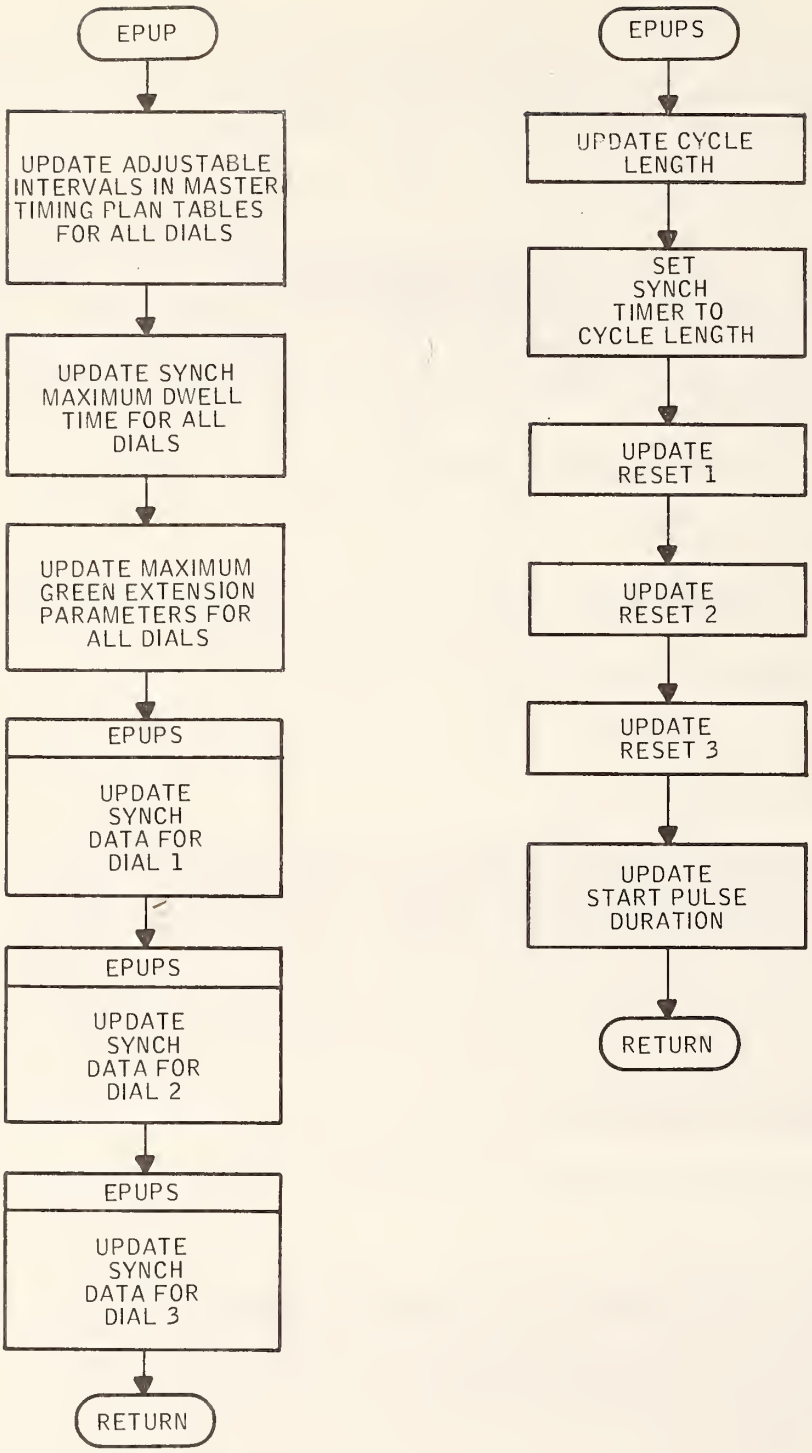


Figure B-22. Update Programming Panel Data Routine (Executive CPC)

3.3.20.7 Test Data

To be determined.

3.3.20.8 Revisions

None.

3.3.21 Convert Binary Coded Digits to Integer Routine (EPBI)

3.3.21.1 Program Narrative

3.3.21.1.1 Purpose

This routine converts the binary-coded digits input from the programming panel thumbwheel switches to integer values.

3.3.21.1.2 Software Interfaces

This routine is referenced via entry point EPBI3 and EPBI2 to convert three and two digit values, respectively when called from routines EPP1, EPP2, and EPP3. At entry, AC0 contains the integer value.

3.3.21.1.3 Assumptions

The raw datum in AC0 is assumed to contain 4-bit codes for each decimal digit arranged as follows from least to most significant bits:

Bits 0-3 Units digit

Bits 4-7 Tens digit

Bits 8-11 Hundreds digit

3.3.21.1.4 Solution Approach

Each four-bit field is as a simple integer value for the corresponding decimal digit. Each value is tested for validity within the range 0 to 9 (binary 0000 to 1001). An out-of-range digit results in an error return to the calling program.

3.3.21.1.5 Program Operation

The digits are converted in the following sequence: hundreds digit, tens digit, units digit. Each four-bit field is extracted from the raw datum and converted to the integer value by means of shifting and adding. The scaling of each digit is implied by the bit positions of the digit in the raw datum. The cumulative converted value is saved in AC2. The raw datum in AC0 is unaltered.

3.3.21.1.6 Model

Not applicable.

3.3.21.1.7 Date: July 1975

3.3.21.1.8 Remarks

None.

3.3.21.2 Program Work Areas

The operations are completed using only the CPU registers, AC0, AC1 and AC2. Local data masks are used for extraction of four-bit fields and testing for valid validity.

3.3.21.3 Program Flowchart

The program flowchart is presented in Figure B-23.

3.3.21.4 Program Listing

The program listing is presented in Appendix D.

3.3.21.5 Input/Output Data Formats

Not applicable.

3.3.21.6 Report Formats

Not applicable.

3.3.21.7 Test Data

To be determined.

3.3.21.8 Revisions

None.

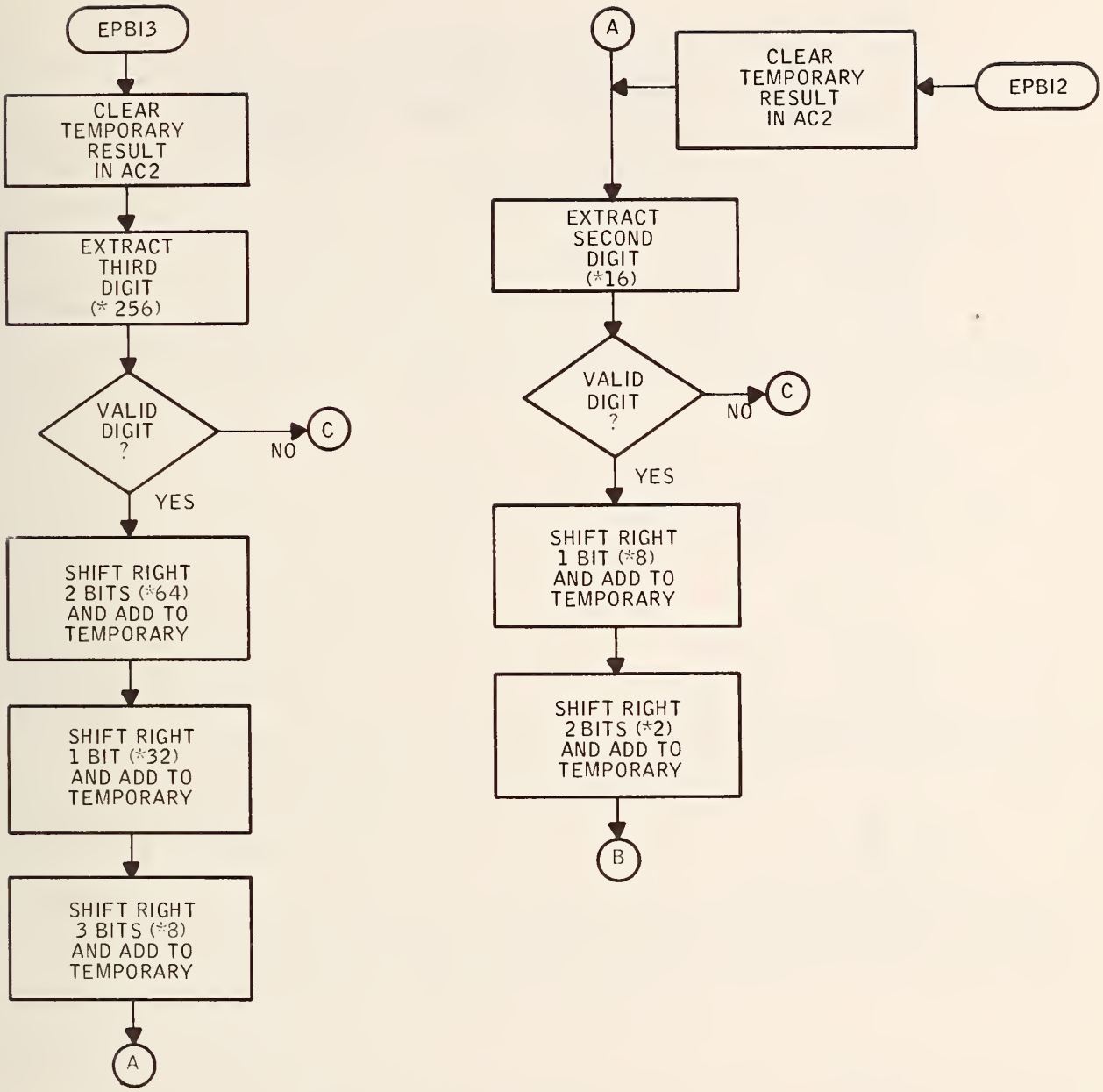


Figure B-23. Convert Binary Coded Digits to Integer Routine (Executive CPC)

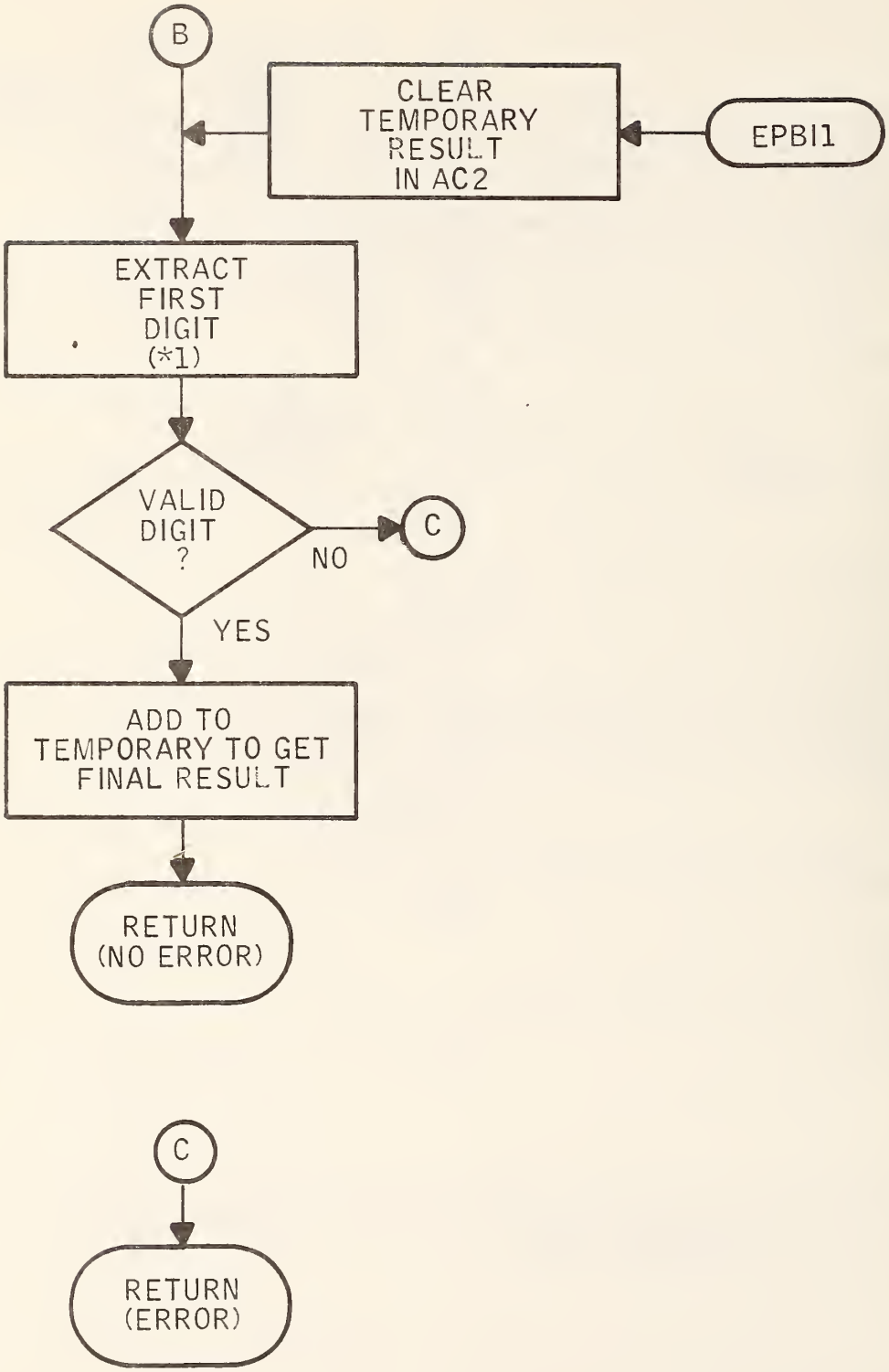


Figure B-23. Convert Binary Coded Digits to Integer Routine (Executive CPC) (Continued)

3.3.22 Routine to Multiply an Integer by 10 (EM10)

This routine multiplies the integer value in AC1 by 10 using shifting and adding. AC0 is used for temporary storage.

The program flowchart is presented in Figure B-24.

The program listing is presented in Appendix D.

3.3.23 Routine to Multiply and Integer by 24 (EM24)

This routine multiplies the integer value in AC1 by 24 using shifting and adding. AC0 is used for temporary storage.

The program flowchart is presented in Figure B-25.

The program listing is presented in Appendix D.

3.3.24 Error Recording Routines (ERE0-ERE12)

3.3.24.1 Program Narrative

3.3.24.1.1 Purpose

These routines store the error data values in the error buffers, set the appropriate error bit in the error flag word and increment the error counter.

3.3.24.1.2 Software Interfaces

These routines are called during initialization or during the RTC sequence by routines that diagnose the error condition. The entry point (ERE0 to ERE12) implies the error condition and dictates where the error data is stored.

3.3.24.1.3 Assumptions

Upon entry, AC1 contains the error datum and AC3 contains the base address of the error buffer. The error buffer is assumed to start with the error flag word and is followed by pairs of locations for storing the error datum and the error counter for each error condition.

3.3.24.1.4 Solution Approach

The program contains thirteen entry points ERE0-ERE12. At each entry point, the error condition is implicit such that

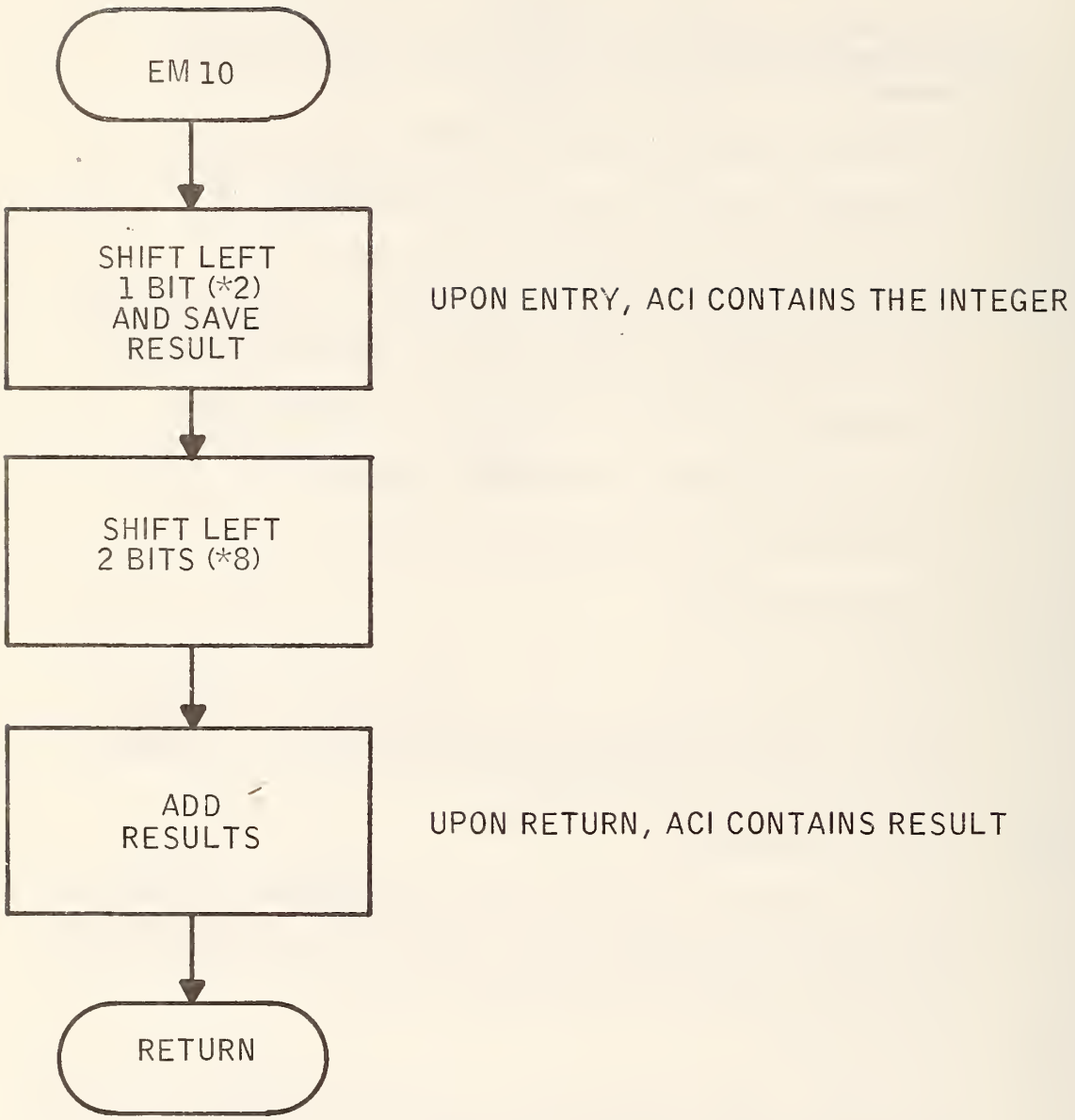


Figure B-24. Routine to Multiply an Integer by 10 (Executive CPC)

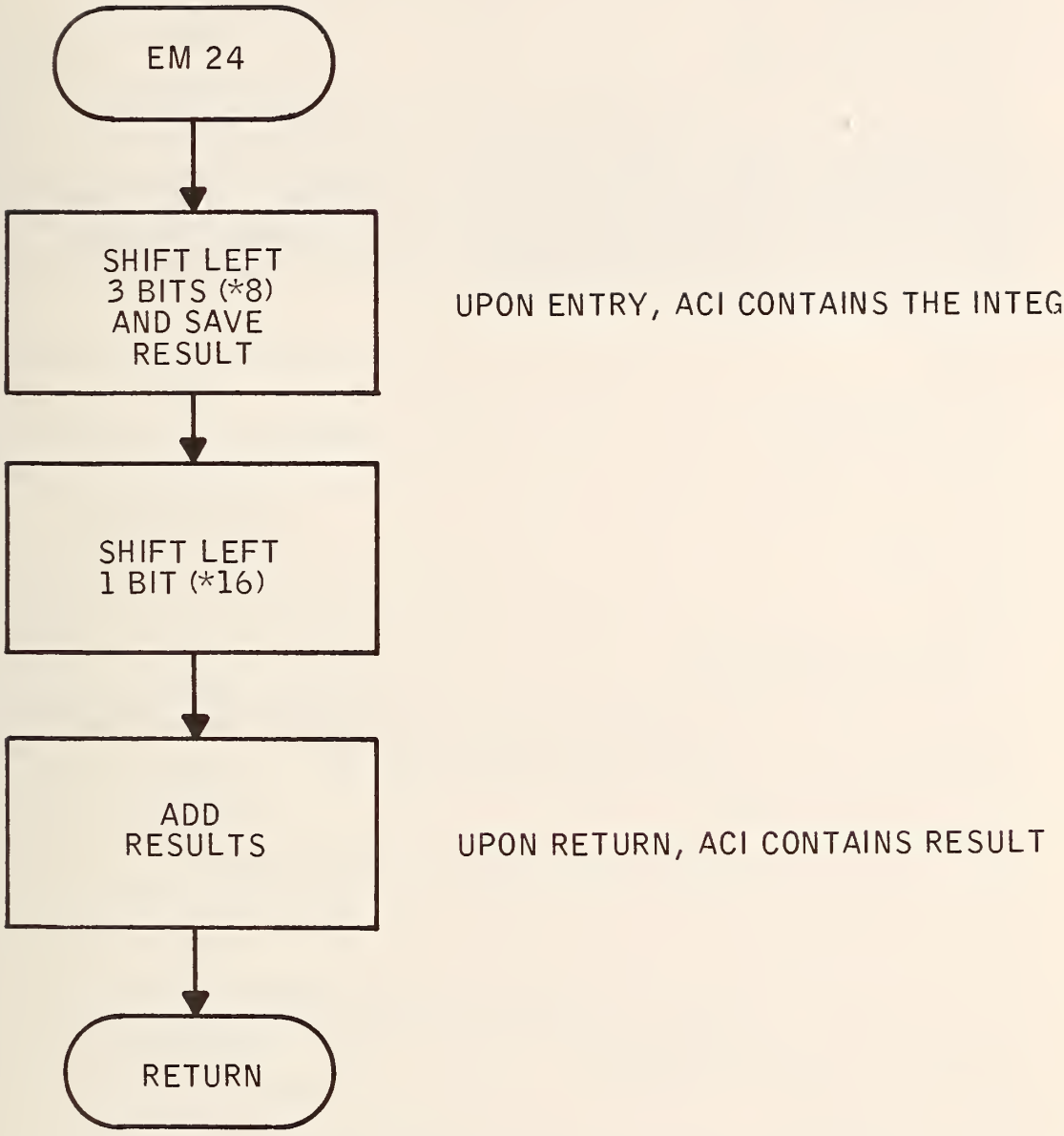


Figure B-25. Routine to Multiply an Integer by 24 (Executive CPC)

the appropriate error flag bit can be set and the error data can be stored in the appropriate pair of locations in the error buffer.

3.3.24.1.5 Program Operation

The operations at each entry point are identical, except for the displacements for the implied error condition. The error datum in AC1 is stored at the appropriate displacement from the base address of the error buffer in AC3. The error counter is then fetched, incremented, and stored. The error flag word is fetched, the appropriate error bit is set, and the flag word is stored.

3.3.24.1.6 Model

Not applicable.

3.3.24.1.7 Date: July 1975

3.3.24.1.8 Remarks

None.

3.3.24.2 Program Work Areas

Data is stored and fetched from the error buffer whose base address is in AC3 upon entry to this routine.

3.3.24.3 Program Flowchart

The program flowchart is presented in Figure B-26.

3.3.24.4 Program Listing

The program listing is presented in Appendix D.

3.3.24.5 Input/Output Data Formats

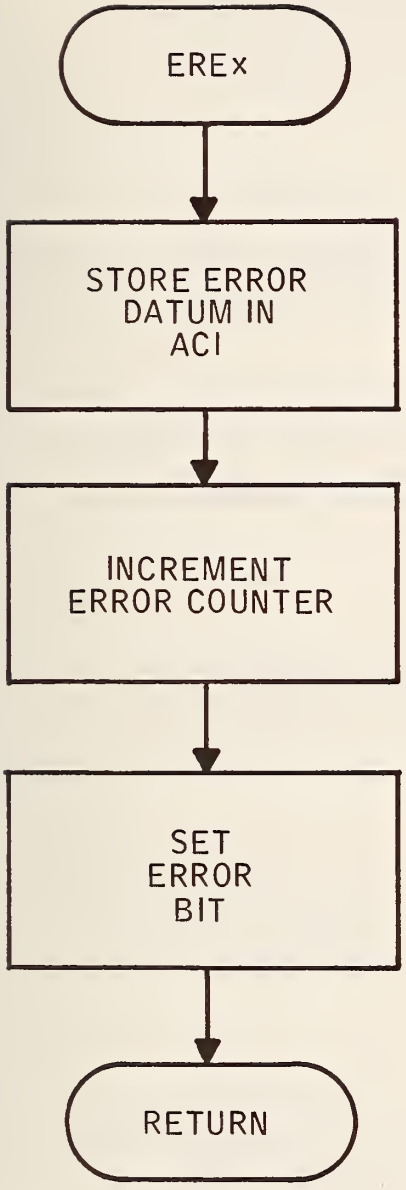
Not applicable.

3.3.24.6 Report Formats

Not applicable

3.3.24.7 Test Data

To be determined.



ENTRY POINTS: ERE0 THROUGH ERE12

ENTRY POINT IMPLIES ERROR BIT IN ERROR FLAG WORD AND DISPLACEMENTS OF ERROR DATUM AND ERROR COUNTER FROM ERROR BUFFER BASE ADDRESS IN AC3 AT ENTRY

Figure B-26. Error Recording Routines (Executive CPC)

3.3.24.8 Revisions

None.

3.3.25 Initialize Computer Control Unit Routine (ECCU)

3.3.25.1 Program Narrative

3.3.25.1.1 Purpose

This routine initializes the RAM data base for the CCU and makes the initial entry into the CCU servicing routine.

3.3.25.1.2 Software Interfaces

This routine is called only during the initialization sequence from ERSI. This routine makes the initial entry to the CCU servicing routine. This routine initializes the RAM data base for the CCU.

3.3.25.1.3 Assumptions

The CCU servicing is assumed to be located at X'EF00 and to begin with the instruction value X'86F7. The CCU data base is assumed to be located in the 48 words starting at address X'ECD0.

3.3.25.1.4 Solution Approach

Tests are made prior to any initialization functions to assure that the CCU is in the system. If the tests indicate that the CCU is not in the system, a return to the calling program is taken immediately.

3.3.25.1.5 Program Operaion

Jump condition X'C is tested with the BOC instruction to test for the CCU in the system. If the jump condition is true, then an additional test on the CCU servicing routine is made. The first instruction is accessed and compared to the local data value. If the instruction agrees, then initialization continues.

The CCU RAM is first cleared to all zeroes. Then the state word is set to 1 and the two breakpoint addresses are set to -1. The return address is initialized to the location of the RTS instruction in this routine, the PUSH/PULL pointer is set to the first location in the RAM work area, and AC0 is set to 0

for the initial display on the CCU. The CCU routine is then entered via a jump through pointer to preclude a stack entry as the return will be made to the address previously stored in the CCU data base.

3.3.25.1.6 Model

Not applicable.

3.3.25.1.7 Date: September 1975

3.3.25.1.8 Remarks

This routine must interface with the CCU servicing routine that is supplied with that device.

3.3.25.2 Program Work Areas

This routine initializes the RAM data base provided for the CCU.

3.3.25.3 Program Flowchart

The program flowchart is presented in Figure B-27.

3.3.25.4 Program Listing

The program listing is presented in Appendix D.

3.3.25.5 Input/Output Data Formats

Not applicable.

3.3.25.6 Report Formats

Not applicable.

3.3.25.7 Test Data

To be determined.

3.3.25.8 Revisions

None.

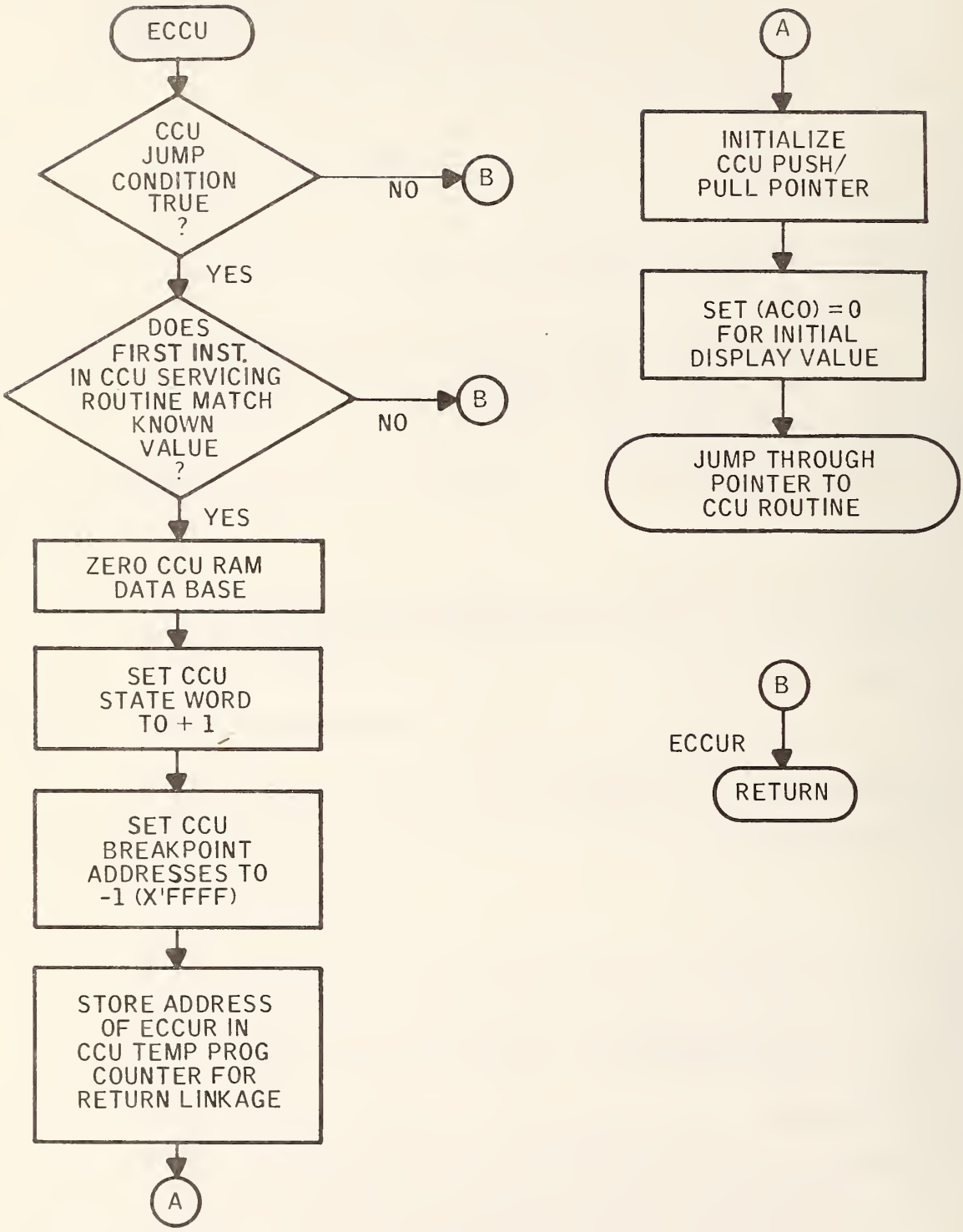


Figure B-27. Initialize Computer Control Unit Routine (Executive CPC)

3.3.26 Initialization RAM Address Test Monitor Routine (EMAD)

3.3.26.1 Program Narrative

3.3.26.1.1 Purpose

This routine monitors the RAM tests performed during the initialization sequence.

3.3.26.1.2 Software Interfaces

This routine is called only by the restart initialization routine, ERSI. This routine calls SMAD to test each block of RAM available for testing. In the event of error, the autodisconnect routine, EADC, is called. Hexadecimal constants from the base page are referenced to establish the start and end addresses for the tests.

3.3.26.1.3 Assumptions

The end of the data base is assumed to be just prior to ADEND. The CCU RAM is assumed to be in the range X'ECD0 to ECFF. The no error return from SMAD is at a nonstandard return displacement of 1 and the error return at the standard displacement of 0.

3.3.26.1.4 Solution Approach

The RAM containing the data base is tested from just beyond the data base (ADEND) to the end of the available RAM. The CCU RAM is tested from the beginning of the page up to the beginning of the CCU data base. Because this program is executed in the initialization sequence, an error results in an immediate call to the autodisconnect routine.

3.3.26.1.5 Program Operation

The RAM containing the data base is tested over the unused locations. The CCU RAM is then tested. An error results in autodisconnect.

3.3.26.1.6 Model

Not applicable.

3.3.26.1.7 Date: September 1975

3.3.26.1.8 Remarks

Refer to routines EBKD and COTF for differences in the RAM test during real time operation.

3.3.26.2 Program Work Areas

This routine initializes the start and end address data for the RAM address test routine in data base locations SSAD and SEAD, respectively.

3.3.26.3 Program Flowchart

The program flowchart is presented in Figure B-28.

3.3.26.4 Program Listing

The program listing is presented in Appendix D.

3.3.26.5 Input/Output Data Formats

Not applicable.

3.3.26.6 Report Formats

Not applicable.

3.3.26.7 Test Data

To be determined.

3.3.26.8 Revisions

None.

3.3.27 RAM Address Test Routine (SMAD)

3.3.27.1 Program Narrative

3.3.27.1.1 Purpose

This routine performs a test on a block of consecutive addresses in RAM.

3.3.27.1.2 Software Interfaces

This routine is called by EMAD during initialization and is called by EBKD during each cycle through the background

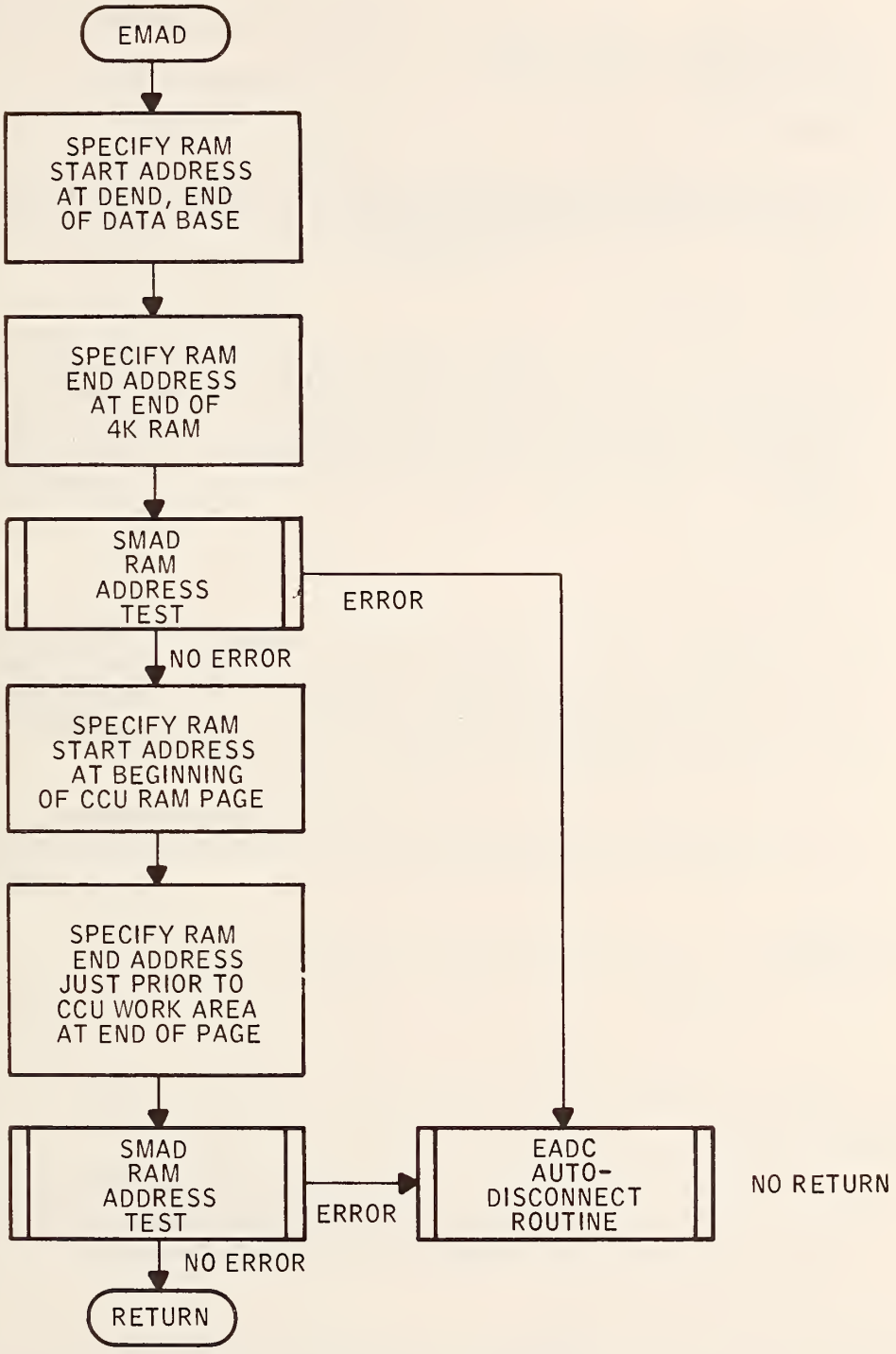


Figure B-28. Initialization RAM Address Test Monitor Routine (Executive CPC)

computations. The start and end addresses for the test are accessed in the data base.

3.3.27.1.3 Assumptions

The block of addresses are assumed to be in RAM. The start and end addresses are assumed to be loaded into the data base prior to entry to this routine.

3.3.27.1.4 Solution Approach

Each location in the RAM address block is tested. Two consecutive errors at the same location are required for stopping the test and taking the error exit. The no-error condition is indicated by a nonstandard return with a displacement of 1. The error return is indicated by the standard return.

3.3.27.1.5 Program Operation

The address value is stored at each location in the specified block of addresses. The contents of each location is then accessed and compared to the address. The first mismatch at a location results in a retry. If the retry also fails, the error exit is taken.

At the error exit, the erroneous contents are loaded into AC2, the address remains in AC3, and CPU status flag 4 is set.

3.3.27.1.6 Model

Not applicable.

3.3.27.1.7 Date: September 1975

3.3.27.1.8 Remarks

For a RAM consisting of discrete parts which provide 1 bit in each memory word over some address range, the address at which this test fails should indicate the bad part.

3.3.27.2 Program Work Areas

The start and end address for the test are found in SSAD and SEAD, respectively. AC0 is used as the consecutive failure counter. CPU status flag 4 is set for an error.

3.3.27.3 Program Flowchart

The program flowchart is presented in Figure B-29.

3.3.27.4 Program Listing

The program listing is presented in Appendix D.

3.3.27.5 Input/Output Data Formats

Not applicable.

3.3.27.6 Report Formats

Not applicable.

3.3.27.7 Test Data

To be determined.

3.3.27.8 Revisions

None.

3.3.28 Catch-on-the-Fly Routine (COTF)

3.3.28.1 Program Narrative

3.3.28.1.1 Purpose

This routine provides a catch-on-the-fly capability for programs executing in the RTC interrupt environment.

3.3.28.1.2 Software Interfaces

This routine is called by the executive scheduling routine, ESCH, conditional upon the jump conditions X'E as tested by the BOC instruction. This routine transfers control to the first location in free RAM where patch code may be entered via the CCU.

3.3.28.1.3 Assumptions

The RTC is assumed to be running at entry to this routine and is restarted prior to exit from this routine. The RAM patch code is assumed to make a standard return to this routine.

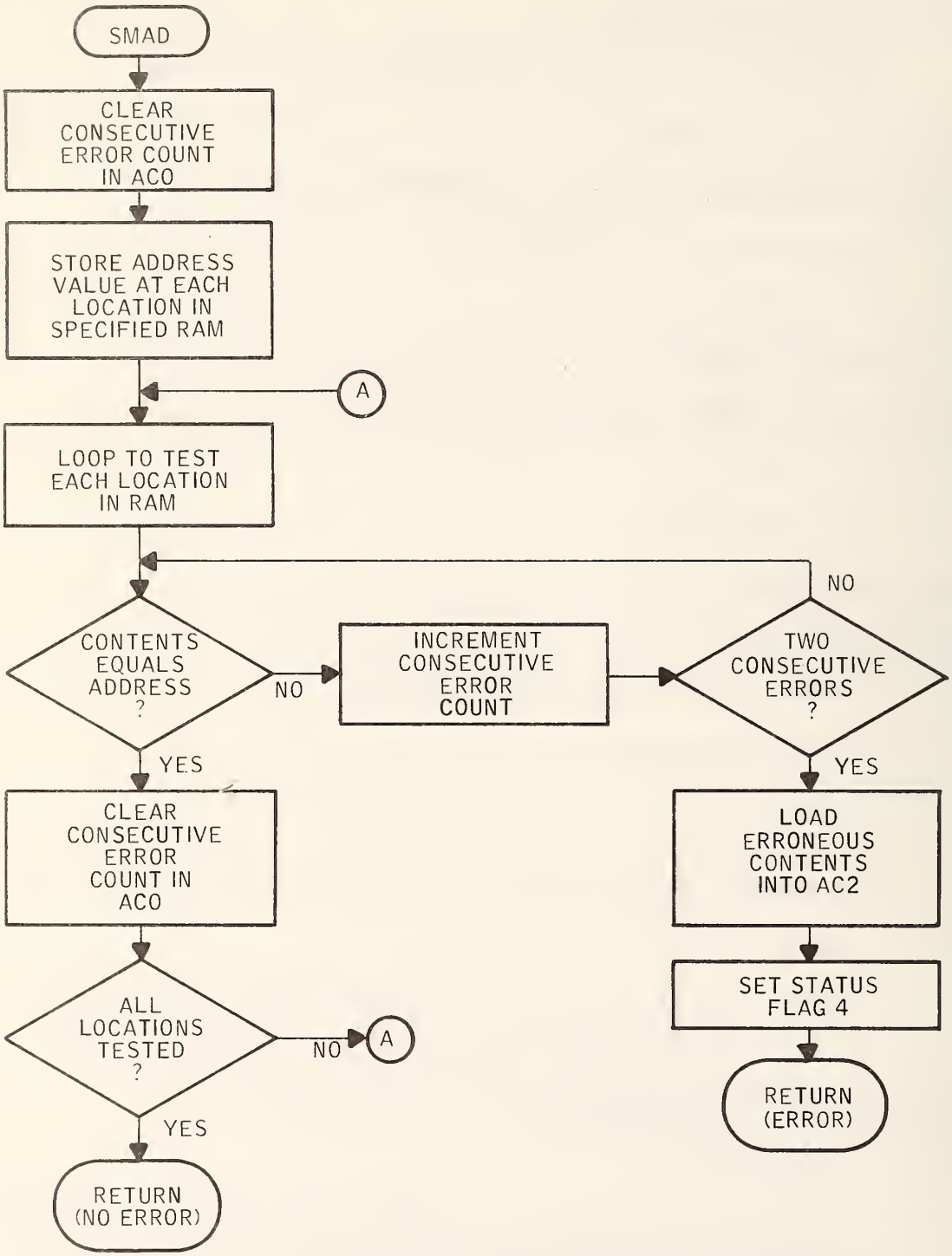


Figure B-29. RAM Address Test Routine (Executive CPC)

3.3.28.1.4 Solution Approach

The computational environment is saved, the RTC disabled, and control transferred to the first location in the free RAM. Following return, the clock is restarted and the environment is restored. Interrupts are not enabled or disabled by this routine.

3.3.28.1.5 Program Operation

The accumulators, RALU status flags, stack and select flag are saved. The RTC is stopped and the interrupt pulse assured to be reset. The transfer is made to the free RAM. The clock is then started and the computational environment restored.

3.3.28.1.6 Model

Not applicable.

3.3.28.1.7 Date: July 1975

3.3.28.1.8 Remarks

For COTF operation in a RAM, locations are reserved by NOP instructions for patching or for the insertion of HALT or SNAP operations with the DEBUG routine.

3.3.28.2 Program Work Areas

The computational environment is saved in the catch-on-the-fly save area, CSAV, in the data base.

3.3.28.3 Program Flowchart

The program flowchart is presented in Figure B-30.

3.3.28.4 Program Listing

The program listing is presented in Appendix D.

3.3.28.5 Output Data Formats

The output control pulses issued by this routine are completely defined by the peripheral device address and function code and are not dependent upon the contents of Accumulator 0.

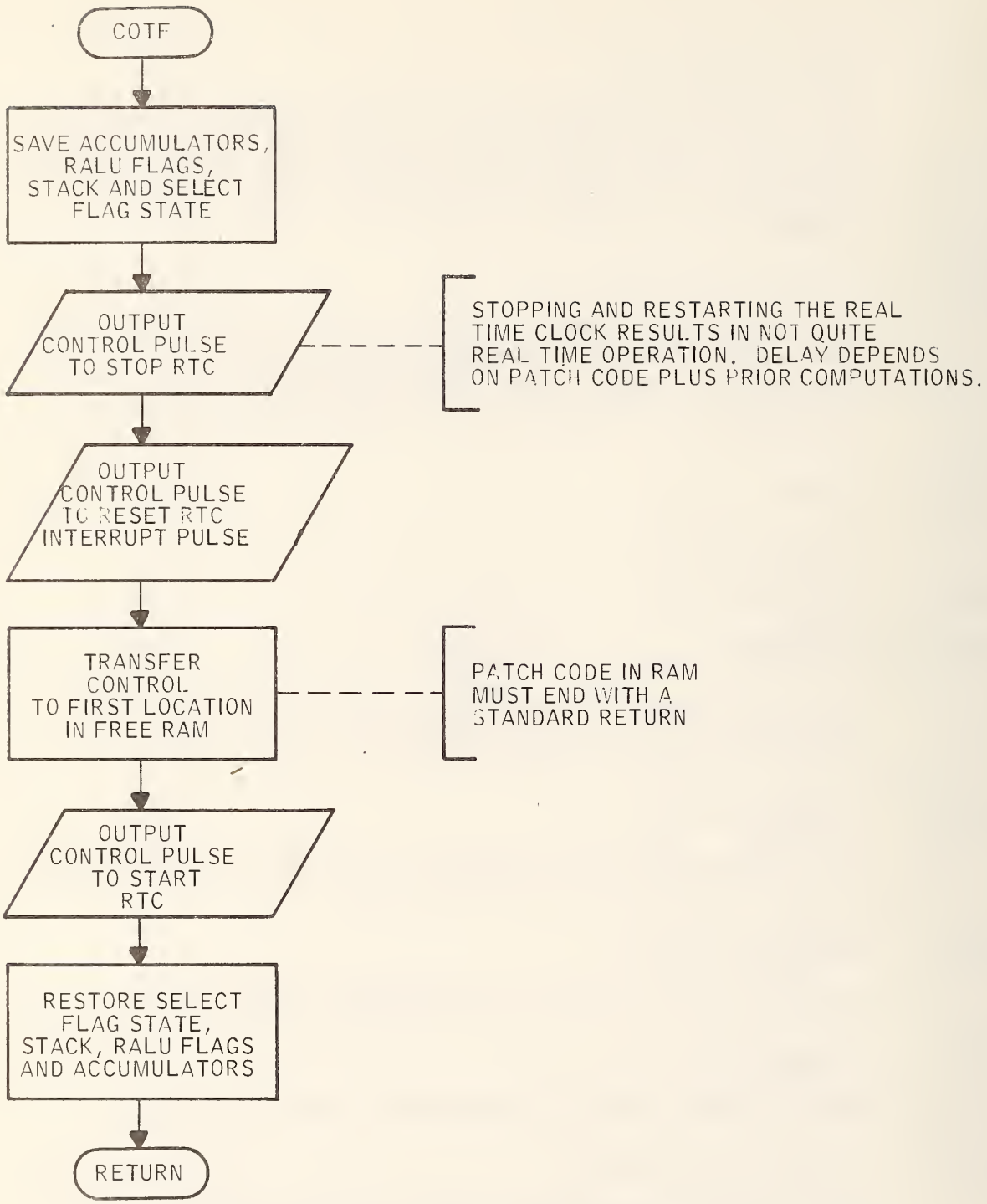


Figure B-30. Catch-on-the-Fly Routine (Executive CPC)

3.3.28.6 Report Formats

Not applicable.

3.3.28.7 Test Data

To be determined.

3.3.28.8 Revisions

None.

3.4 DATA INPUT CPC

This computer program component contains the routines that perform the program controlled input from peripheral devices.

3.4.1 Data Input for Real Time Control Sequence, BRTC

3.4.1.1 Program Narrative

3.4.1.1.1 Purpose

This routine inputs the control bits from the transfer control panel and loop master controller, inputs the binary switch bits from the programming panel, and inputs the signal display command registers for main street and cross street signal commands.

3.4.1.1.2 Software Interfaces

This routine is called only by the executive scheduling routine, ESCH, during each real time clock interval. Refer to Table B-2 for the data fetched and the data stored by this routine.

3.4.1.1.3 Assumptions

Not applicable.

3.4.1.1.4 Solution Approach

The program sets up the peripheral device addresses in the appropriate register and executes the data input instruction that contains the input function code. Each datum is input to the MCU via Accumulator 0 and is stored in the data input portion of the data base. BBUF contains the control bit discrete input word and the command registers. BPPD contains the programming panel switch word.

3.4.1.1.5 Program Operation

The data items are input to Accumulator 0 and are stored in the data base in a straightforward sequence.

3.4.1.1.6 Model

Not applicable.

3.4.1.1.7 Date: July 1975

3.4.1.1.8 Remarks

None.

3.4.1.2 Program Work Areas

The discrete input word containing the control bits from the TCP and the LMC is stored in BDIN, the first location in BBUF; the command register for main street is stored in BCR1, the second location in BBUF; and the command register for cross street is stored in BCR2, the third location in BBUF. The programming panel switch word is stored in BBPD.

3.4.1.3 Program Flowchart

The program flowchart is presented in Figure B-31.

3.4.1.4 Program Listing

The program listing is presented in Appendix D.

3.4.1.5 Input Data Formats

The control bits in the discrete input word are defined in Table B-3.

The bits in the command register word for main street are defined in Table B-4.

The bits in the command register word for cross street are defined in Table B-5.

The bits in the programming panel switch word are defined in Table B-6.

3.4.1.6 Report Formats

Not applicable.

3.4.1.7 Test Data

To be determined.

3.4.1.8 Revisions

None.

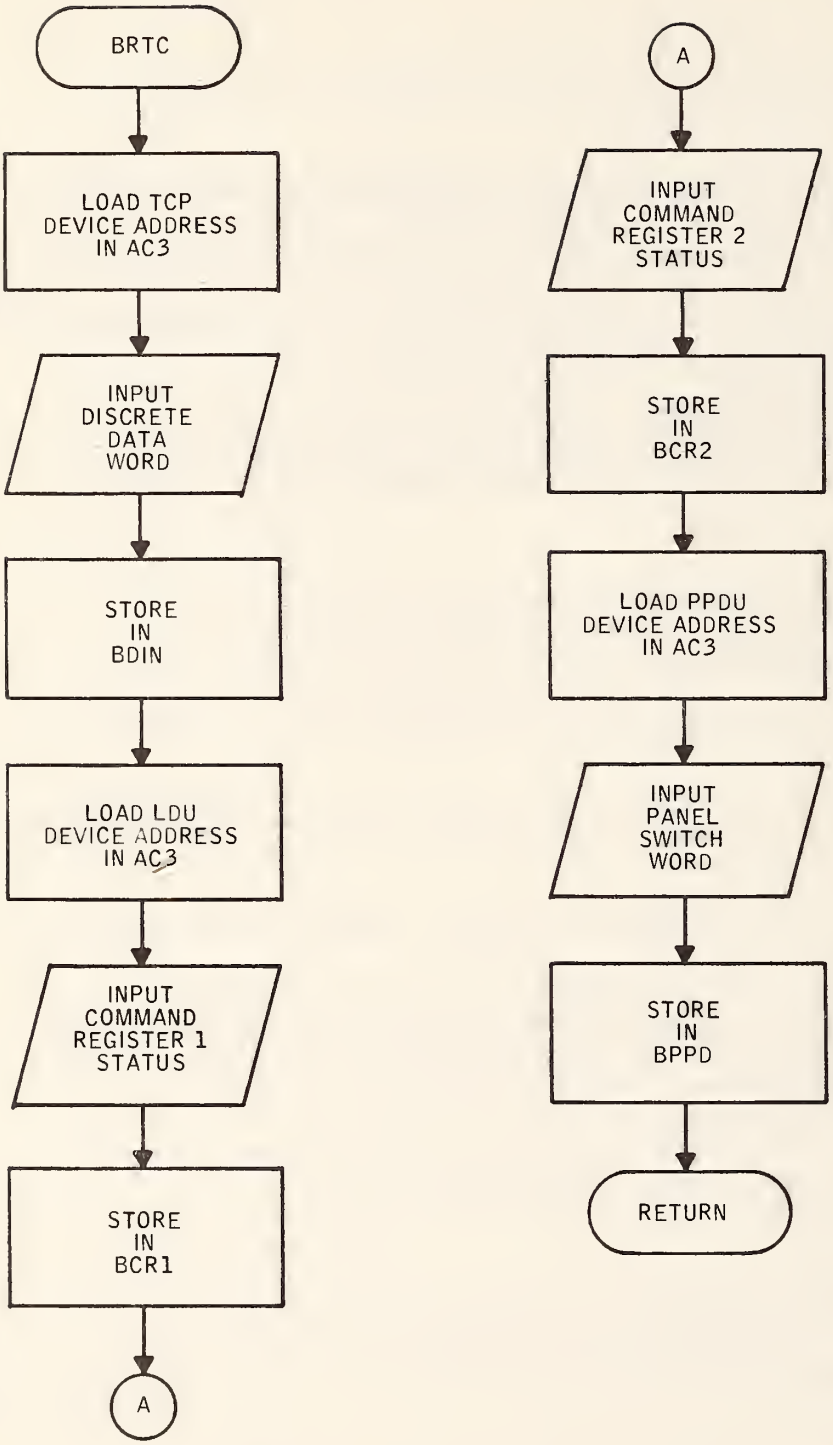


Figure B-31. Data Input for Real Time Control Sequence Routine (Data Input CPC)

3.4.2 Data Input for Programming Panel Thumbwheel Switches Routine (BPAN)

3.4.2.1 Program Narrative

3.4.2.1.1 Purpose

This routine inputs the data words containing the binary code for the digits displayed on the thumbwheel switches of the programming panel. All thumbwheel switch data are input during the execution of this program.

3.4.2.1.2 Software Interfaces

This routine is called by EPAN during the restart initialization sequence and by EPPM during the RTC interval in which the INPUT request pushbutton on the panel is first depressed. The thumbwheel switch data are input from three peripheral devices PP1U, PP2U and PP3U and are stored in the data base arrays BPD1, BPD2, and BPD3 for the three dials, respectively. This routine calls BPIN to input the thumbwheel data from one of the three peripheral devices.

3.4.2.1.3 Assumptions

The raw data arrays, BPD1, BPD2, BPD3 are assumed to directly follow BPPD in the data base. Each raw data array is assumed to be exactly 7 elements in length.

3.4.2.1.4 Solution Approach

The same function code is used to input the corresponding parameter from each of the three dials on the programming panel. The base address for the first dial is loaded into an index register. The peripheral device is loaded into the proper register, and subroutine BPIN is called to sequence through the function codes, to input the data one by one, and to store each datum in the data base. Between calls to BPIN, the base address in the data base is incremented by the length of the raw data input buffer.

3.4.2.1.5 Program Operation

The data are input to Accumulator 0 and stored in the data base one by one. The thumbwheel data is read in for dial 1, then dial 2, and then dial 3.

3.4.2.1.6 Model

Not applicable.

3.4.2.1.7 Date: July 1975

3.4.2.1.8 Remarks

None.

3.4.2.2 Program Work Areas

This routine stores the thumbwheel switch data in the following data base buffers:

BPD1 - Dial 1 data

BPD2 - Dial 2 data

BPD3 - Dial 3 data

The sequence of data in each input buffer is listed in Table B-7.

3.4.2.3 Program Flowchart

The program flowchart is presented in Figure B-32.

3.4.2.4 Program Listing

The program listing is presented in Appendix D.

3.4.2.5 Input Data Formats

The formats for the data words containing the binary coded thumbwheel digits are defined in Table B-11.

3.4.2.6 Report Formats

Not applicable.

3.4.2.7 Test Data

To be determined.

3.4.2.8 Revisions

None.

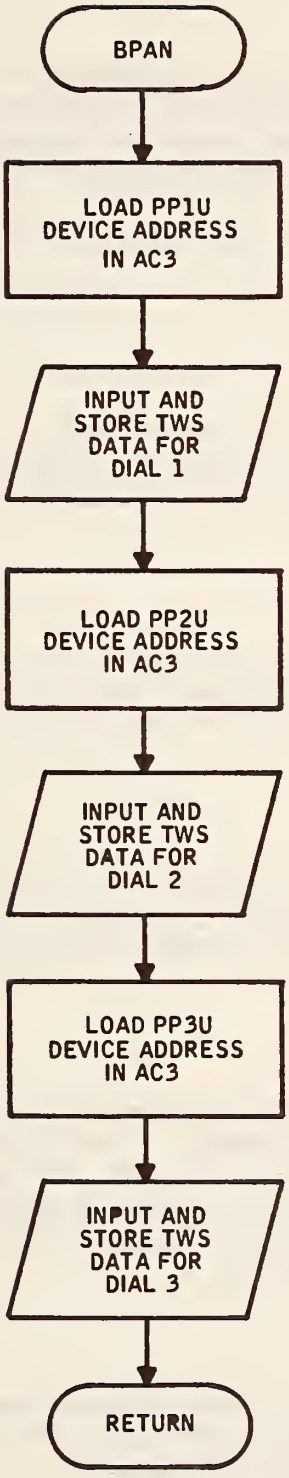


Figure B-32. Data Input for Programming Panel Thumbwheel Switches Routine (Data Input CPC)

3.5 DATA OUTPUT CPC

This computer program component contains the routines that format output data and perform the program controlled output to peripheral devices. Some routines in other CPC's issue output control pulses that require no data value.

3.5.1 Data Output for Real Time Control Sequence Routine (XRTC)

3.5.1.1 Program Narrative

3.5.1.1.1 Purpose

This routine outputs the main street and cross street command words to the LDU to drive the signal lights and outputs the status word to the SDU to drive the status display lights.

3.5.1.1.2 Software Interfaces

This routine is called only by the executive scheduling routine, ESCH, during each real time clock interval. Refer to Table B-2 for the data fetched and the data stored by this routine.

3.5.1.1.3 Assumptions

The status display output word and the main street and cross street command output words are fetched from the first, second, and third locations, respectively, in the XBUF array in the data base.

3.5.1.1.4 Solution Approach

The base address of XBUF is loaded into an index register for addressing the output words. The program then sets up the peripheral device address for the LDU and outputs the main street and cross street command output words. The peripheral device address for the SDU is then set up and the status display word is output.

3.5.1.1.5 Program Operation

The data items are loaded from the data base into Accumulator 0 and then are output to the appropriate peripheral device in a straightforward sequence.

3.5.1.1.6 Model

Not applicable.

3.5.1.1.7 Date: July, 1975

3.5.1.1.8 Remarks

None.

3.5.1.2 Program Work Areas

The output words are fetched from the XBUF array in the data base.

3.5.1.3 Program Flowchart

The program flowchart is presented in Figure B-33.

3.5.1.4 Program Listing

The program listing is presented in Appendix D.

3.5.1.5 Output Data Formats

The formats of the main street and cross street output command words are defined in Tables B-9 and B-10 respectively.

The format of the status display word is defined in Table B-8.

3.5.1.6 Report Formats

Not applicable.

3.5.1.7 Test Data

To be determined.

3.5.1.8 Revisions

None.

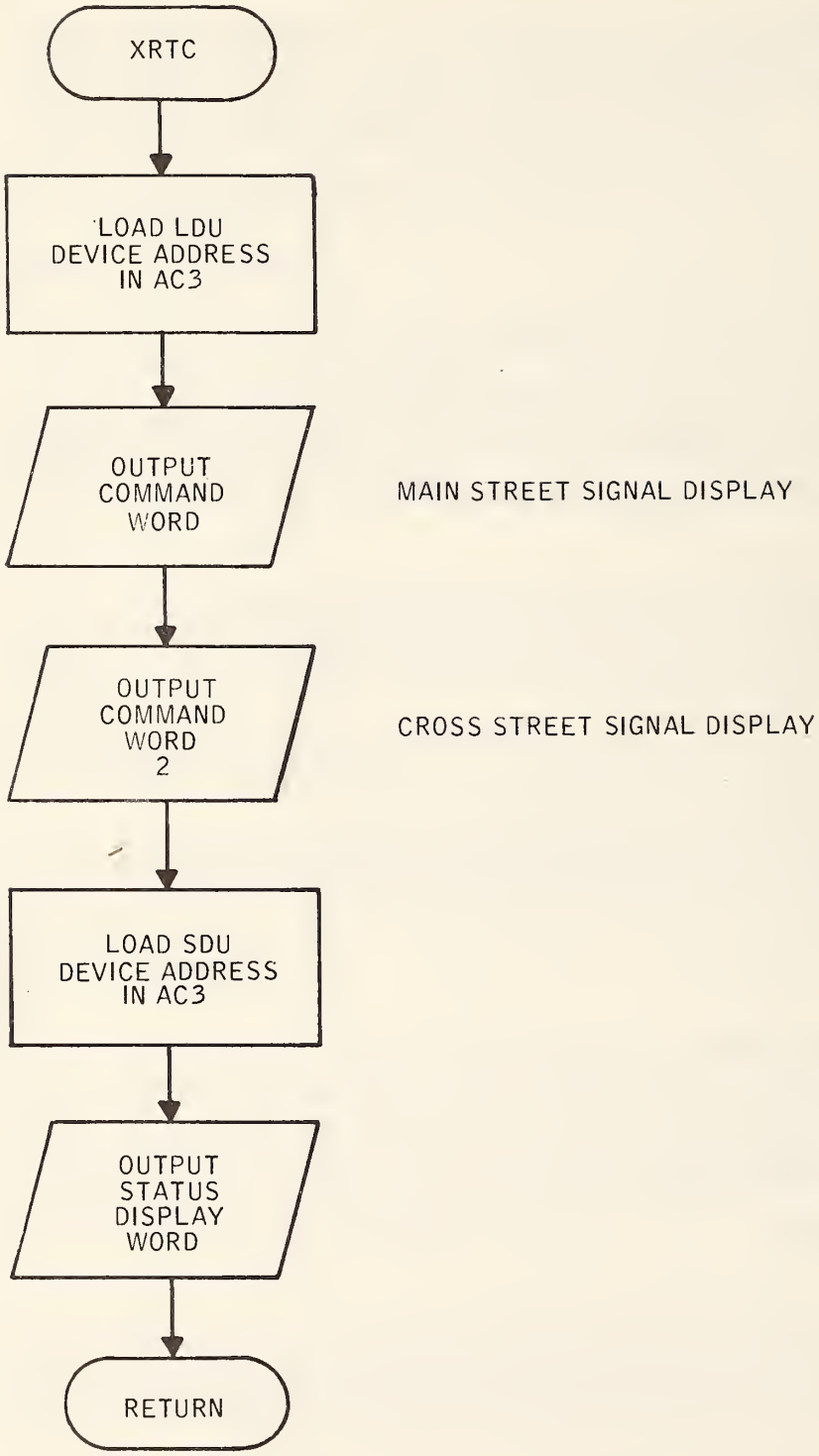


Figure B-33. Data Output for Real Time Control Sequence Routine (Data Output CPC)

3.5.2 Reset Watchdog Timer Routine (XWDT)

3.5.2.1 Purpose

This routine issues the control pulse to reset the watchdog timer in the SDMU.

3.5.2.1.2 Software Interfaces

This routine is called only by the command monitoring routine, ECOM, upon satisfaction of the command monitoring conditions.

3.5.2.1.3 Assumptions

None.

3.5.2.1.4 Solution Approach

The peripheral device address for the watchdog timer is set up and the output instruction with the reset control pulse function code is executed.

3.5.2.1.5 Program Operation

Refer to program listing.

3.5.2.1.6 Model

Not applicable.

3.5.2.1.7 Date: July, 1975

3.5.2.1.8 Remarks

None.

3.5.2.2 Program Work Areas

None.

3.5.2.3 Program Flowchart

The program flowchart is presented in Figure B-34.

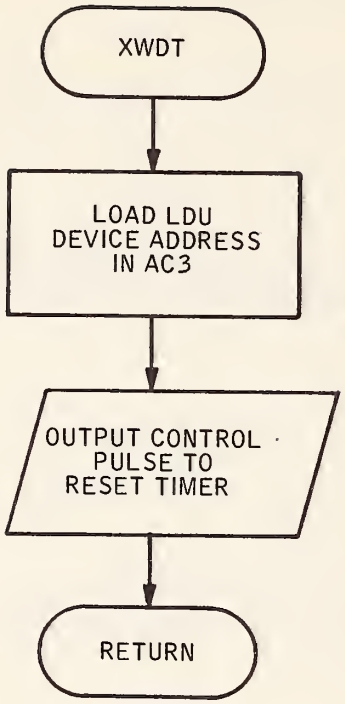


Figure B-34. Reset Watchdog Timer Routine (Data Output CPC)

3.5.2.4 Program Listing

The program listing is presented in Appendix D.

3.5.2.5 Output Data Formats

The watchdog timer peripheral device responds to the device address/function code without regard to the data contained in Accumulator 0.

3.5.2.6 Report Formats.

Not applicable.

3.5.2.7 Test Data

To be determined.

3.5.2.8 Revisions

None.

3.5.3 Format of the Signal Display Output Command Words Routine (XSDC)

3.5.3.1 Program Narrative

3.5.3.1.1 Purpose

This routine constructs the output command words for the signal display on main street and cross street. The update bits in the command words are set or reset as appropriate.

3.5.3.1.2 Software Interfaces

This routine is called by the executive scheduling routine, ESCH, during each real time clock interval following execution of the timing plan execution routine, TPEX. The command index word, TCIWC, contains the indices for both commands. The executive error flag for command feedback is tested. Command index errors are recorded in the Data Output CPC error buffer, XERR.

3.5.3.1.3 Assumptions

The main street command word is output prior to the cross street command words.

3.5.3.1.4 Solution Approach

A command format table contains the bit patterns for the commands to be issued to main street or cross street. The table is located in PROM. The command index word specified by the Timing Plan Execution CPC contains the indices for both main street and cross street command formats. Bit 15 in the output command word is the update bit which, when set, causes parallel update of all circuit driving bits for both main street and cross street from the output registers to the command registers in the LDU.

3.5.3.1.5 Program Operations

The command index word is first tested for a new command issued by the Timing Plan Execution CPC at the beginning of a new interval. For a new command, the output command words are formatted and the update bit is set in the cross street command word. In the absence of a new command, the error flags for command feedback are tested for an error condition. If an error has occurred, the previous command

words with the update bit set in the cross street command shall be formatted to update the command registers and to clear a one-shot command error. In the absence of a new command and command feedback error, the update bits are cleared in the output command words. As each command word is formatted, the command index is checked against the extent of the command format table. The command indices refer to the location in the table of command formats.

3.5.3.1.6 Model

Not applicable.

3.5.3.1.7 Date: July, 1975

3.5.3.1.8 Remarks

None.

3.5.3.2 Program Work Areas

The feedback command error for the current cycle is ECMF. Non-zero value indicates an error. Command index error data is stored in the XERR buffer. The new command index word is TCIWC. The new command index word is copied to XCIWC (current command) and XCIWC is copied to XCIWP (previous command) at the start of each new command interval. The command format table is XCFT.

3.5.3.3 Program Flowchart

The program flowchart is presented in Figure B-35.

3.5.3.4 Program Listing

The program listing is presented in Appendix D.

3.5.3.5 Output Data Formats

Not applicable.

3.5.3.6 Report Formats

Not applicable.

3.5.3.7 Test Data

To be determined.

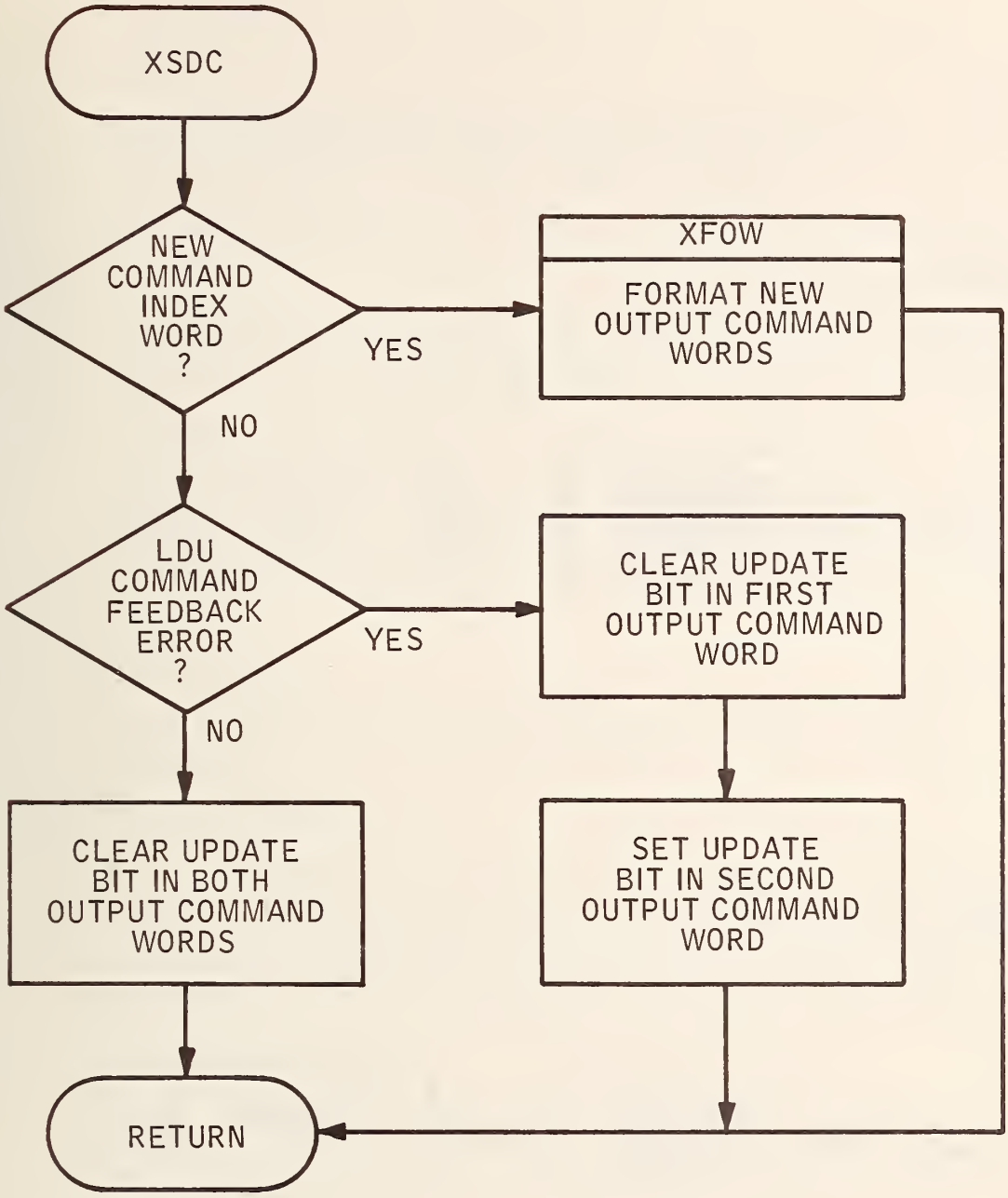


Figure B-35. Format Signal Display Output Command Words Routine (Data Output CPC)

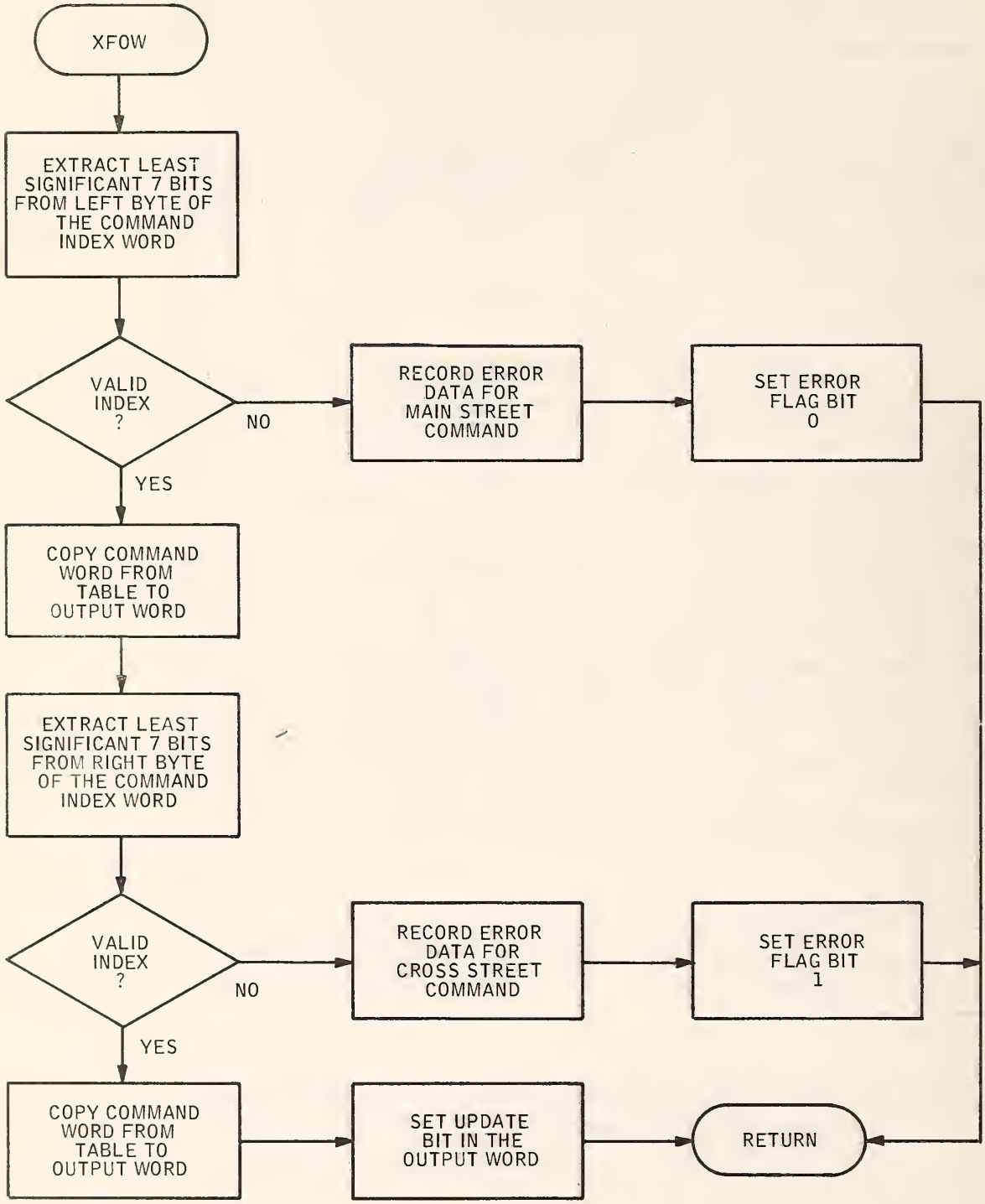


Figure B-35. Format Signal Display Output Command Words Routine (Data Output CPC) (Concluded)

3.5.3.8 Revisions

None.

3.5.4 Turn Programming Panel Status Lights On and Off Routines (XLON1, XLOF1, XLON2, XLOF2)

3.5.4.1 Program Narrative

3.5.4.1.1 Purpose

This routine issues the output control pulses to the status lights on the programming panel.

3.5.4.1.2 Software Interfaces

This routine is entered at one of the four entry points for the single purpose of turning a status light ON or OFF. Calls are made by EPAN during initialization, by EPPM during real time operation, and by TNCY at update of the programming panel data.

3.5.4.1.3 Assumptions

The peripheral device address is set up by the calling routine.

3.5.4.1.4 Solution Approach

The peripheral device address is loaded into the appropriate register by the calling routine. The data register is cleared and the output instruction with the appropriate function code is executed. The program can then then be used for all peripheral devices constituting the programming panel.

3.5.4.1.5 Program Operation

Each entry point results in the same program sequence except for the function code. Refer to the program listing.

3.5.4.1.6 Model

Not applicable.

3.5.4.1.7 Date: July, 1975

3.5.4.1.8 Remarks

None.

3.5.4.2 Program Work Areas

None.

3.5.4.3 Program Flowchart

The program flowchart is presented in Figure B-36.

3.5.4.4 Program Listing

The program listing is presented in Appendix D.

3.5.4.5 Output Data Format

Not applicable.

3.5.4.6 Report Formats

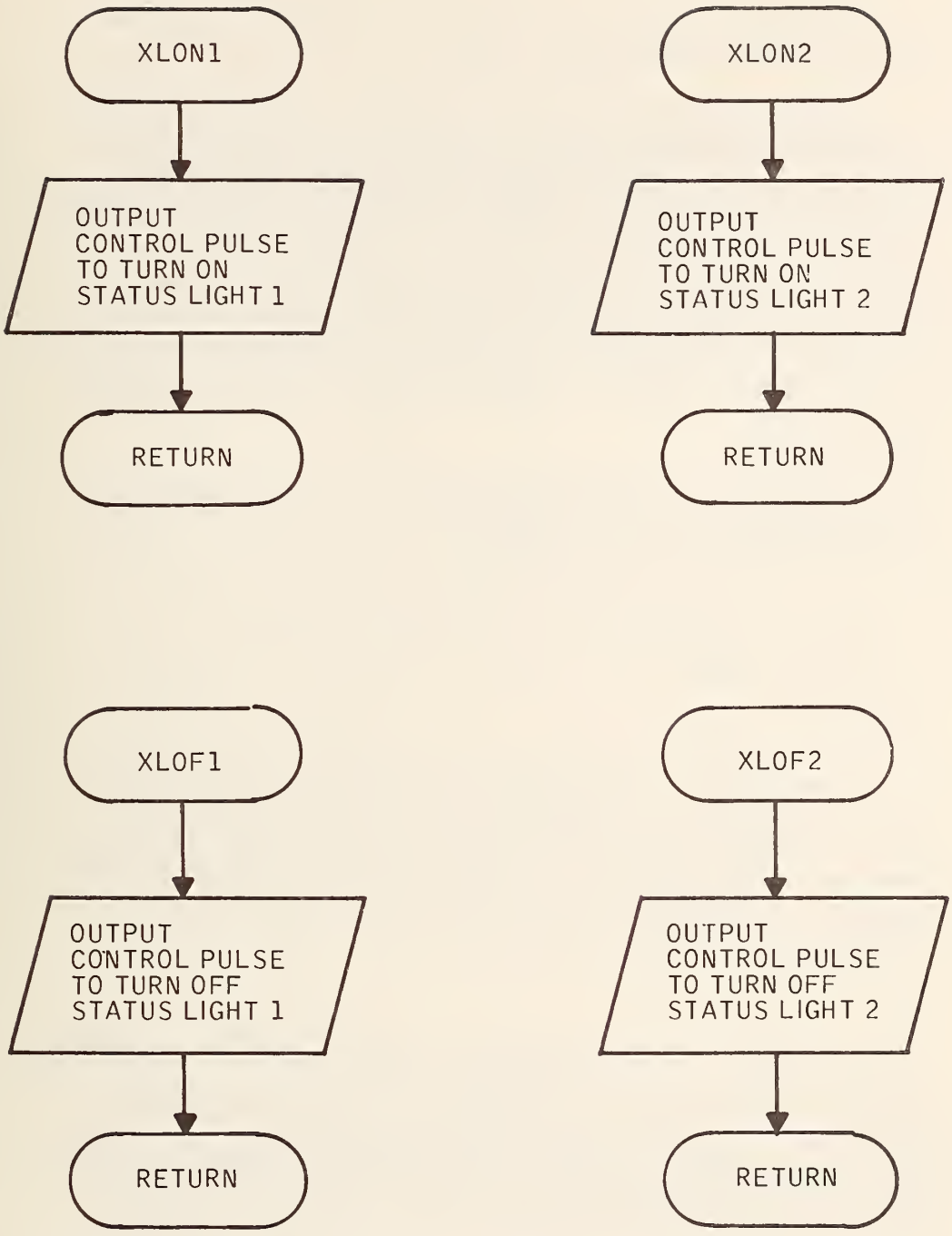
Not applicable.

3.5.4.7 Test Data

To be determined.

3.5.4.8 Revisions

None.



NOTE: AT ENTRY, AC3 CONTAINS PERIPHERAL DEVICE ADDRESS FOR PP1U, PP2U, PP3U OR PPDU

Figure B-36. Turn Programming Panel Status Lights On and Off Routines (Data Output CPC)

3.6 TIMING PLAN EXECUTION CPC

The routines in the timing plan execution computer program component perform the functions of timing the sequence of intervals in the timing plan, specifying the signal display commands to the output routine, transferring from one plan to another, dwelling for synchronization with the loop master controller and dwelling for transfer of control to and from the local controller. Data from the read/write memory are accessed via address pointers in the base page of memory. Local data constants are stored in the read only memory with the programs in the CPC. Refer to Table B-2 for the data items set and referenced by the programs in the timing plan execution CPC.

The timing plan is represented by two parallel tables, the interval durations, and the signal display command indices for main street and cross street.

3.6.1 Timing Plan Execution Monitor Routine (TPEX)

3.6.1.1 Program Narrative

3.6.1.1.1 Purpose

This routine times the sequence of intervals in the timing plan and controls dwell for synchronization with the loop master controller and dwell for transfer of control to and from the local controller. TPEX indirectly controls signal displays, dial changes, and synchronization via calls to subordinate routines.

3.6.1.1.2 Software Interfaces

The timing plan execution monitor routine, TPEX, is called only by the executive scheduling routine, ESCH, during each real time clock interval. TPEX calls the timing plan synchronized start routine, TSST, at each real time clock interval during the first timing plan interval (MSG). TPEX calls the next interval routine, TNXT, at the end of each timing plan interval.

3.6.1.1.3 Assumptions

The timing plan interval data is valued in units of the 0.1-second real time clock interval. The timing plan begins with the first interval in the main street green sequence and

is currently limited to twenty intervals. The end of the timing plan is followed by an interval with a negative duration. The intervals are timed sequentially. No intervals are skipped. Zero duration intervals are permitted.

A bus preempt in progress precludes a dial transfer (change of timing plan) and precludes synchronization dwell. Synchronization dwell is allowed only at the beginning of the timing plan. Maximum synchronization dwell is 25 percent of the cycle length.

Three timing plans are available. All timing plans have the same structure and sequence of signal displays. All fixed interval durations are identical for the three timing plans.

3.6.1.1.4 Solution Approach

The timing plan is represented by two parallel tables, the interval durations and the signal display command indices for main street and cross street. At the beginning of an interval, the current interval timer, TINT, is initialized to the interval duration and the command index word is transferred to the output routine via the data base. These operations are done by routines subordinate to TPEX. The timing plan is initialized to the transfer hold interval and dwells there until the discrete input changes from the WAIT state to the CONTROL state. At each pass through the transfer hold interval, the wait bit of the discrete input word is tested. If the wait state is present, timing plan execution will dwell until the state changes.

Timing plan interval duration tables are maintained for both the current and the next cycle of the timing plan to incorporate the bus preempt modification capability. The preempt functions may modify interval durations in one or both of the timing plan tables.

3.6.1.1.5 Program Operation

The timing plan execution routine (TPEX) checks to see if the current interval is the transfer hold interval. If it is, and the wait bit of the discrete input word is set, the command index word is copied into the current command index word buffer (TCIWC) and control is returned to the executive scheduler. Otherwise, TPEX checks if the current interval is the first interval of the MSG. If it is, the synchronized start routine (TSST) is called. Otherwise, the time remaining in the timing

interval is decremented. If the time remaining is zero, the next interval routine (TNXT) is called. Otherwise control is returned to the executive scheduler.

3.6.1.1.6 Model

Not applicable.

3.6.1.1.7 Date and Programmer

July 1975, N.G. Planer

3.6.1.1.8 Remarks

None.

3.6.1.2 Program Work Areas

The switches, counters, accumulators, and hold areas referenced or altered by TPEX are defined in Section 3.2.2 or 3.6.6. Table B-2 is a list of the data stored and the data fetched by this routine.

3.6.1.2.1 Switches

BDIN - The discrete input word is referenced through a pointer, ABDIN, in the base page to test the state of bit 14, the transfer hold bit.

3.6.1.2.2 Counters and Accumulators

TIN - The current timing interval index, TIN, is referenced through a pointer, ATIN, in the base page.

TINT - The current interval timer, TINT, is referenced through a pointer, ATINT, in the base page, then altered (decremented), and finally referenced again.

3.6.1.2.3 Hold Areas, Buffers

TCIWC - The current command index word buffer is set via a pointer, ATCIWC, in the base page in the case of transfer hold.

THI - The transfer hold interval index is referenced via a pointer, ATHI, in the base page.

TCIW - The command index word table is referenced via a pointer, ATCIWC, in the base page.

3.6.1.3 Program Flowchart

The program flowchart is presented in Figure B-37.

3.6.1.4 Program Listing

The program listing is presented in Appendix D.

3.6.1.5 Input Data Formats

Not applicable.

3.6.1.6 Report Formats

Not applicable.

3.6.1.7 Test Data

To be determined.

3.6.1.8 Revisions

None.

3.6.2 Synchronized Start Routine (TSST)

3.6.2.1 Program Narrative

3.6.2.1.1 Purpose

This routine causes timing plan execution to dwell in the first interval of the Main Street Green (MSG) sequence until synchronization with the loop master controller can be achieved or until a maximum dwell has been reached. It also sets a flag that communicates to other CPC's that timing has been released from synchronization dwell.

3.6.2.1.2 Software Interfaces

TSST is called by TPEX during each real time clock interval during the first interval in the MSG sequence.

3.6.2.1.3 Assumptions

Refer to the assumption for routine TPEX.

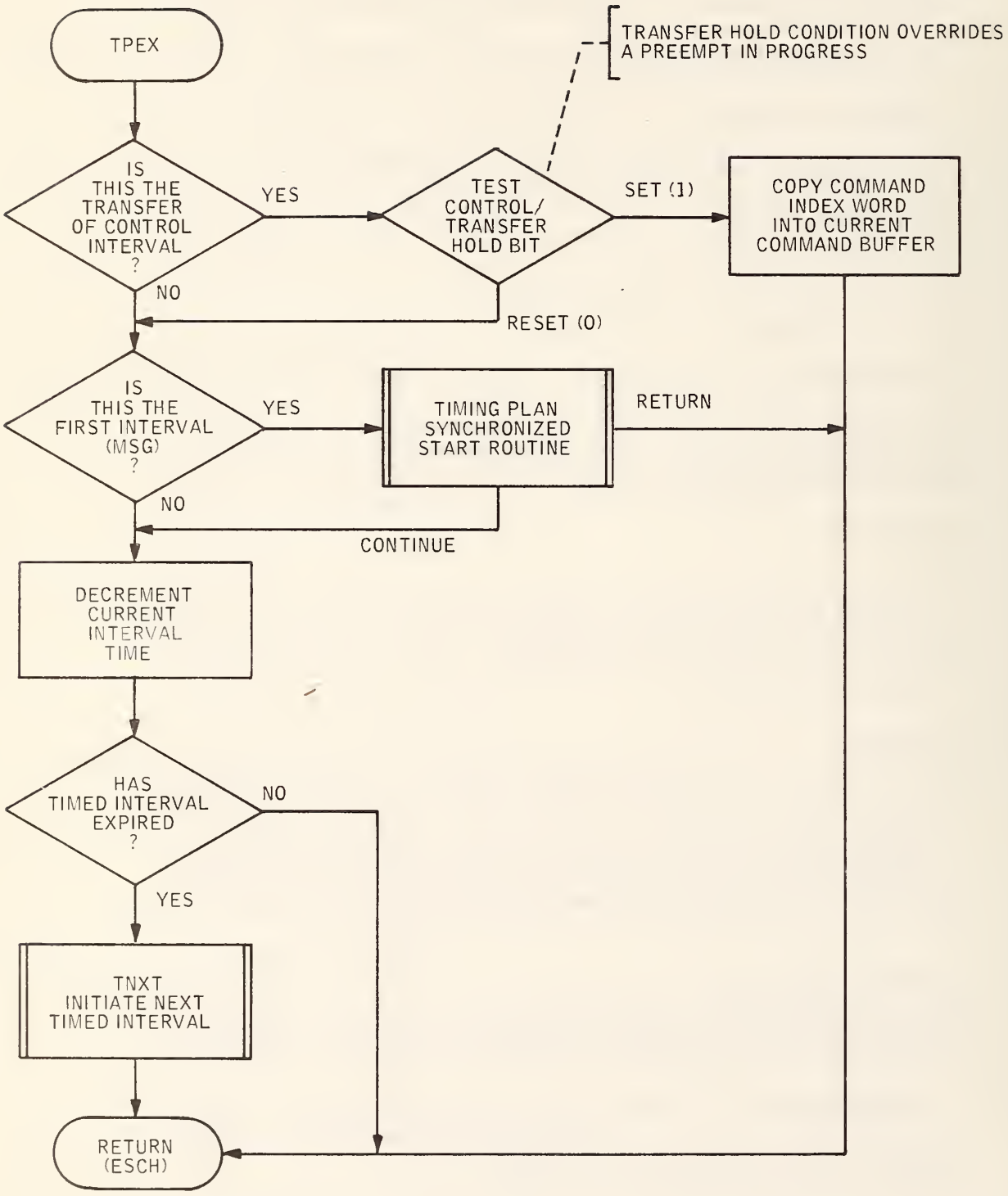


Figure B-37. Timing Plan Execution Monitor Routine (Timing Plan Execution CPC)

3.6.2.1.4 Solution Approach

TSST references a flag (TFST) indicating whether synchronization with the loop master controller is desired. If the value of TFST is less than 1, bit 15 is set (negative value) as an indication that timing of the MSG sequence has begun. If the value of the flag is 1, a timer for maximum dwell is tested. If maximum dwell has been exceeded, bit 15 is set. The start pulse EPST, is tested if the maximum dwell has not been exceeded. If EPST has a value of 1, the start pulse is present and bit 15 of TFST is set. If EPST is zero, dwell continues via a return to the Executive Scheduler routine through TPEX.

3.6.2.1.5 Program Operation

The synchronized start routine sets the release bit (TFST, bit 15) if the "wait for synchronized start" flag (TFST, bit 0) is not set, if the "wait for synchronized start" flag is set and the dwell timer (ETST) has exceeded the maximum dwell for the appropriate dial (TDWL), or if the wait for start flag is set and the start pulse is present (EPST=1). If the release bit has been set by TSST, timing is started or continued. If the release bit has not been set, timing is delayed for one RTC interval (0.1 sec).

3.6.2.1.6 Model

Not applicable.

3.6.2.1.7 Date and Programmer

July 1975, N.G. Planer

3.6.2.1.8 Remarks

None.

3.6.2.2 Program Work Areas

The switches, counters, accumulators and hold areas referenced or altered by TSST are defined in Section 3.2.2 or 3.6.6. Table B-2 is a list of the data stored and the data fetched by this routine.

3.6.2.2.1 Switches

TFST - The wait for synchronized start flag is referenced and set through a pointer, ATFST, in the base page.

TCDS - The current dial selection indicator is referenced by TSST through a pointer, ATCDS, in the base page.

EPST - The start pulse indicator is referenced via a pointer, AEPST, in the base page.

3.6.2.2.2 Counters and Accumulators

ETST - The dwell timer is referenced via a pointer, AETST, in the base page.

3.6.2.2.3 Hold Areas, Buffers

TDWL - The maximum dwell buffer is referenced via a pointer, ATDWL, in the base page.

3.6.2.3 Program Flowchart

The program flowchart is presented in Figure B-38.

3.6.2.4 Program Listing

The program listing is presented in Appendix D.

3.6.2.5 Input Data Formats

Not applicable.

3.6.2.6 Report Formats

Not applicable.

3.6.2.7 Test Data

To be determined.

3.6.2.8 Revisions

None.

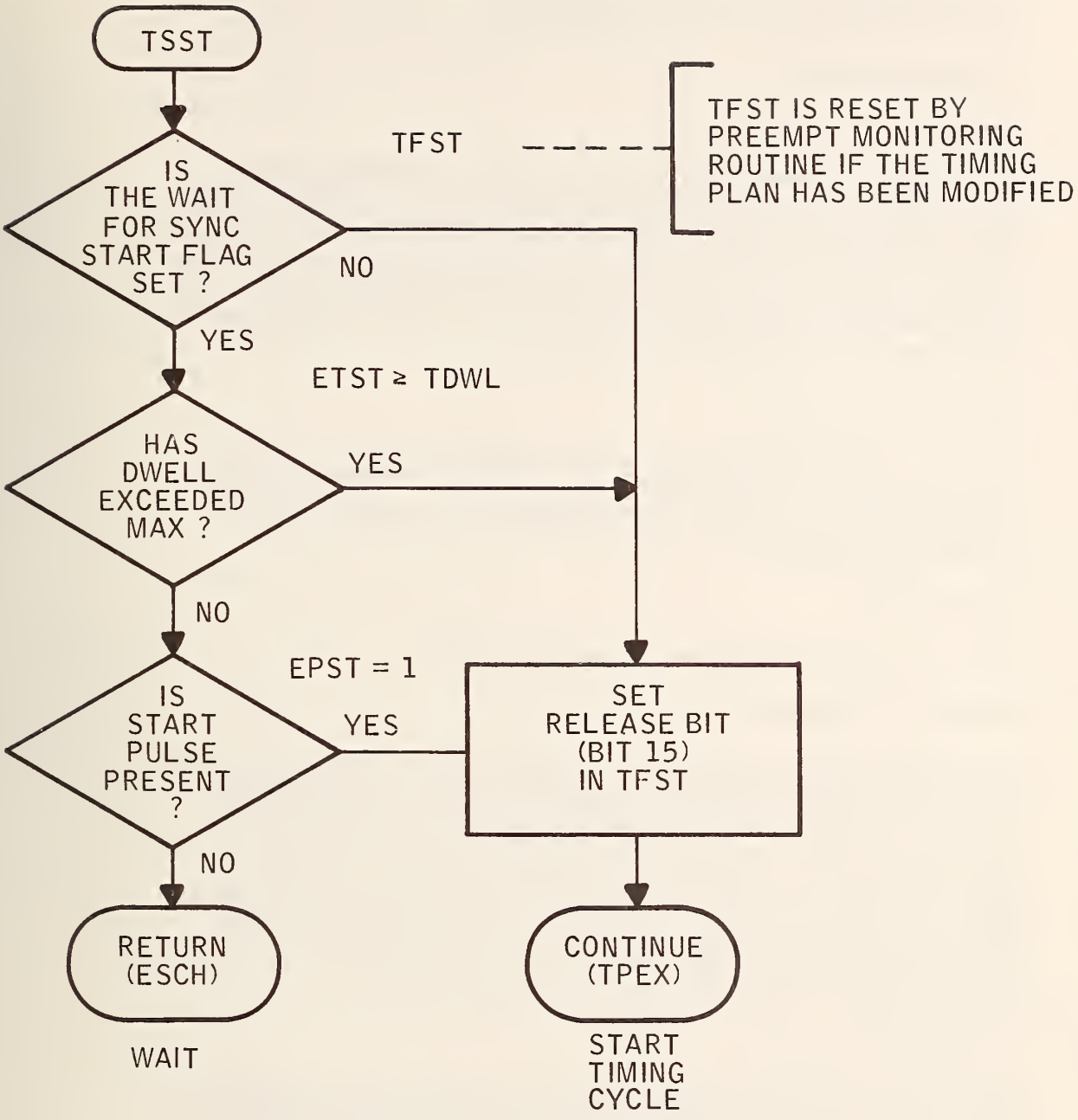


Figure B-38. Synchronized Start Routine (Timing Plan Execution CPC)

3.6.3 Initiate Next Interval Routine (TNXT)

3.6.3.1 Program Narrative

3.6.3.1.1 Purpose

This routine initiates the next interval in the timing plan by advancing the timing interval index at the end of each timing interval, initializing the current interval timer, and copying the appropriate command index word into the current command buffer. TNXT determines when the end of cycle occurs. It also tests the timing interval index for validity and causes an error exit if the index is out of limits.

3.6.3.1.2 Software Interfaces

TNXT is called only by the timing plan execution monitor, TPEX, at the end of each timing interval. TNXT calls the executive end of cycle routine, EEOC, at the end of each timing plan cycle.

3.6.3.1.3 Assumptions

Refer to the assumptions for routine TPEX.

3.6.3.1.4 Solution Approach

This routine increments the current interval index, initializes the current interval timer, and copies the command index word for the new interval into the current command buffer. If the new interval length is less than zero, end of cycle is signaled and the end of cycle routine is called. If the new interval length is zero, a branch to the beginning of TNXT is executed, except in the case of the transfer hold interval. The timing plan will dwell in a zero or non-zero transfer hold interval if the control bit is in the WAIT state.

3.6.3.1.5 Program Operation

The next timing plan interval routine (TNXT) increments the current interval index (TIN) and tests if it is out of limits. If it is out of limits, the autodisconnect routine (EADC) is called. Otherwise, the length of the new interval is tested. If it is less than zero, the index (TIN) is zeroed and the end of cycle routine (EEOC) is called. If the interval length is zero, and transfer hold mode is not indicated, the index (TIN) is incremented again via a jump to the entry point for TNXT.

If transfer hold is indicated, the current interval timer (TINT) is initialized and the new command index word is copied into the buffer (TCIWC).

3.6.3.1.6 Model

Not applicable.

3.6.3.1.7 Date and Programmer

July 1975, NG Planer

3.6.3.1.8 Remarks

None.

3.6.3.2 Program Work Areas

The switches, counters, accumulators, and hold areas referenced or altered by TNXT are defined in Section 3.2.2 or 3.6.6. Table B-2 is a listing of the data stored and data fetched by this routine.

3.6.3.2.1 Switches

BDIN - The discrete input word is referenced via a pointer, ABDIN, in the base page.

TERRF - The error flag word. Bit 1 is set by TNXT if the interval index is out of limits.

3.6.3.2.2 Counters and Accumulators

TIN - The current timing interval index is referenced and altered through a pointer, ATIN, in the base page.

TINT - The current interval timer, TINT, is referenced and altered through a pointer, ATINT, in the base page.

TCIW - The command index word table is referenced via a pointer, ATCIW, in the base page.

TCIWC - The current command index word is altered by TNXT via a pointer, ATCIWC, in the base page.

THI - The transfer hold interval index is referenced by TNXT via a pointer, ATHI, in the base page.

TIMX - The maximum value for the current interval index (TIN) is referenced by TNXT via a pointer, ATIMX, in the base page.

TCTP - The current timing plan table is referenced by TNXT via a pointer, ATCTP, in the base page.

TERR - The timing plan error buffer is altered by TNXT via a pointer, ATERR, in the base page, if the interval index is out of limits.

3.6.3.3 Program Flowchart

The program flowchart is presented in Figure B-39.

3.6.3.4 Program Listing

The program listing is presented in Appendix D.

3.6.3.5 Input Data Formats

Not applicable.

3.6.3.6 Report Formats

Not applicable.

3.6.3.7 Test Data

To be determined.

3.6.3.8 Revisions

None.

3.6.4 Initiate Next Cycle Routine (TNCY)

3.6.4.1 Program Narrative

3.6.4.1.1 Purpose

TNCY performs or initiates the functions required to set up the data base for each new cycle: it resets the "wait for synchronized start" flag; it calls the programmable panel update routine (EPUP), as appropriate; it resets the panel update flag; and it determines whether a new dial selection has been made. In the case of a penal update or a dial change,

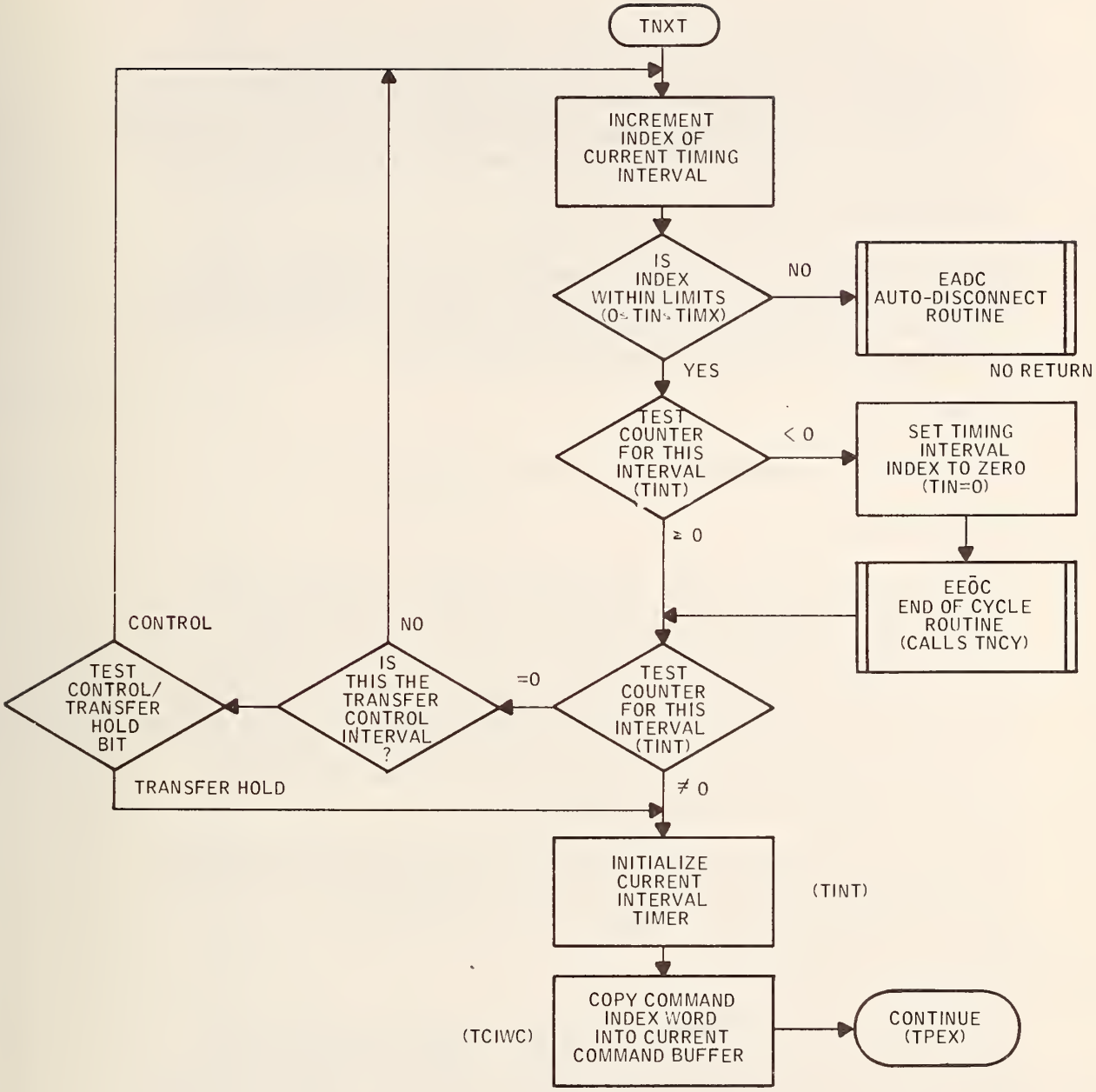


Figure B-39. Initiate Next Interval Routine (Timing Plan Execution CPC)

TNCY initiates the transcription of the appropriate master timing plan to the current and next timing plan buffers (TCTP and TNTP). Otherwise, the next timing plan (TNTP) is copied into the current timing plan buffer (TCTP) and the appropriate master timing plan is copied into the next timing plan buffer.

3.6.4.1.2 Software Interfaces

TNCY is called only by the Executive End of Cycle routine, EEOC, at the end of each controller cycle. TNCY calls the Programmable Panel Update routine, EPUP, if no preempt is in progress and the panel update flag is set.

3.6.4.1.3 Assumptions

Refer to the assumptions for routine TPEX.

3.6.4.1.4 Solution Approach

TNCY tests the preempt in progress flag, the panel synchronization switches, the panel update flag, and the dial selection to determine whether to initiate a panel update and to determine which timing plans to copy. A panel update occurs if the preempt in progress flag is not set but the update flag is set. A preempt in progress precludes a dial transfer. In the case of a dial transfer or a panel update, both the current and next timing plan tables are updated from the master plan for the appropriate dial.

If there is no panel update and no dial transfer, the current timing plan is updated from the next timing plan table, and the next timing plan is updated from the master.

3.6.4.1.5 Program Operation

TNCY is called by the end-of-cycle routine (EEOC) to test the preempt in progress flag (PIP). If PIP is non-zero (preempt in progress), the TCOPY routine is called to copy the next timing plan into the current timing plan buffer and the appropriate master plan into the next timing plan buffer. A return from subroutine TNCY is then executed. Otherwise, the manual override switch is tested. If it is not set, and the appropriate panel synchronization is on, the preempt in progress flag is tested again. If the manual override switch is set, or the appropriate synch switch is not set, the "wait for synch start" flag is reset to zero prior to testing PIP.

If PIP is in non-zero or the panel update flag (EPUF) is not set, the call to EPUP is bypass and the dial selection is determined. If the dial has not changed, TCOPY is called to copy the next timing plan into the current timing plan buffer and the appropriate master into the next timing plan buffer. In the case of a dial change, the new master is copied into both the current and next timing plan buffers. A return from subroutine TNCY is then executed. If there is no preempt in progress and the update flag is set, EPUP is called, the pending input request light on the panel is turned off, and the panel update flag is reset to zero. The appropriate master timing plan is copied into the current and next timing plan buffers via TCOPY, and a return from TNCY is executed.

3.6.4.1.6 Model

Not applicable.

3.6.4.1.7 Date and Programmer

July 1975, N.G. Planer

3.6.4.1.8 Remarks

None.

3.6.4.2 Program Work Areas

3.6.4.2.1 Switches

PIP - The preempt in progress flag is referenced by TNCY via a pointer, APIP, in the base page.

BPPD - The programmable panel discrete switch word is referenced by TNCY via a pointer, ABPPD, in the base page.

TCDS - The current dial selection indicator is referenced and altered by TNCY via a pointer, ATCDS, in the base page.

TFST - The "wait for synchronized start" flag is altered (reset) by TNCY via pointer, ATFST, in the base page.

EPUF - The panel update flag is referenced and altered by TNCY via a pointer, AEPUF, in the base page.

ED23 - The dial select word is referenced by TNCY via a pointer, AED23, in the base page.

3.6.4.2.2 Counters and Accumulators

None.

3.6.4.2.3 Hold Areas, Buffers

TCTP - The current timing plan buffer is referenced by TNCY via a pointer, ATCTP, in the base page.

TNTP - The next timing plan buffer is referenced by TNCY via a pointer, ATNTP, in the base page.

TPM1, TPM2, TPM3 - The master timing plan buffers are referenced by TNCY via address pointers located relative to pointer, ATPP, in the base page.

3.6.4.3 Program Flowchart

The program flowchart is presented in Figure B-40.

3.6.4.4 Program Listing

The program listing is presented in Appendix D.

3.6.4.5 Input Data Formats

Not applicable.

3.6.4.6 Report Formats

Not applicable.

3.6.4.7 Test Data

To be determined.

3.6.4.8 Revisions

None.

3.6.5 Copy Timing Plan Tables Routine (TCOPY)

3.6.5.1 Program Narrative

3.6.5.1.1 Purpose

This routine copies the master timing plan tables into the next and current timing plan buffers, and the next into the current timing plan buffer.

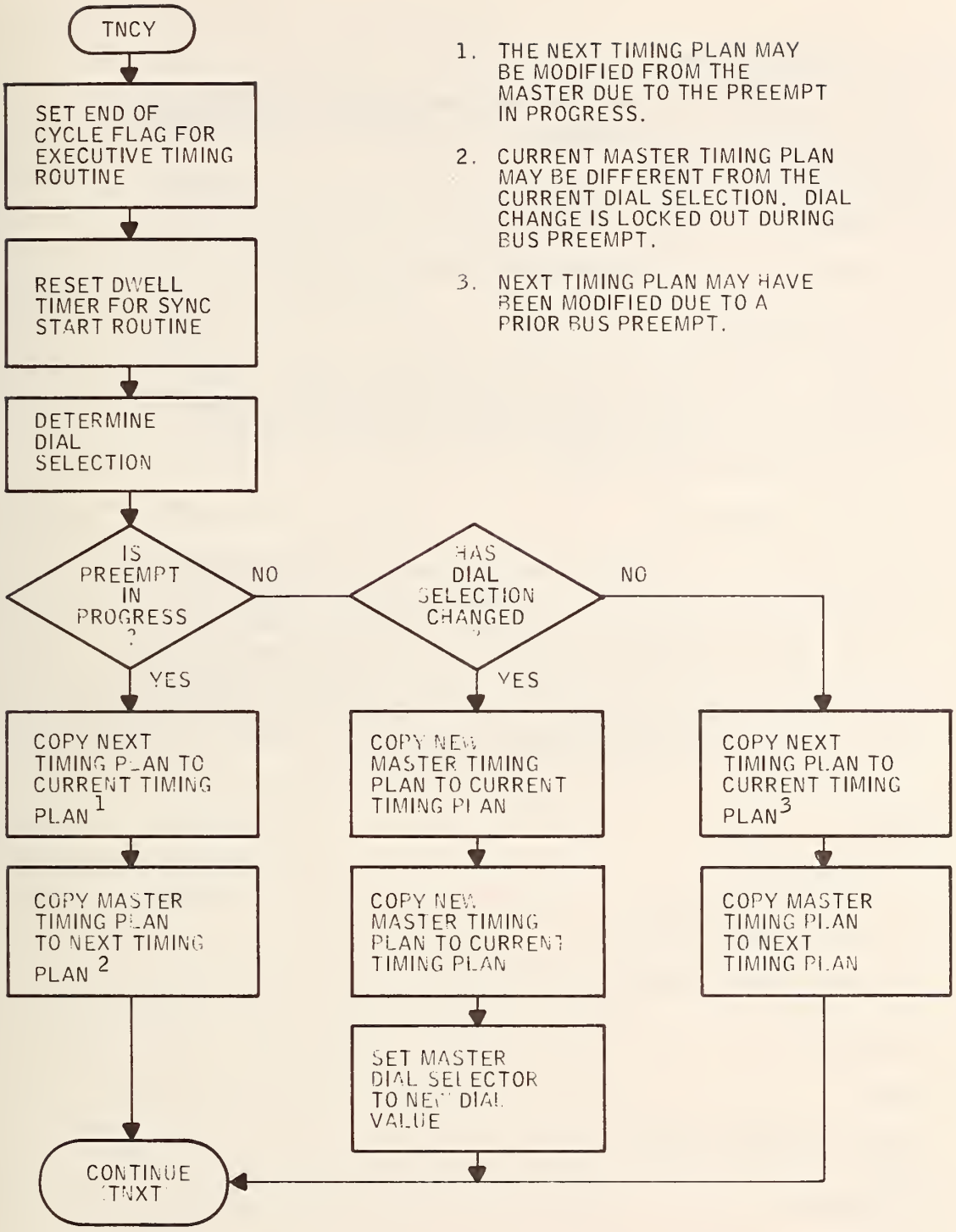


Figure B-40. Initiate Next Cycle Routine (Timing Plan Execution CPC)

3.6.5.1.2 Software Interfaces

TCOPY is called only by TNCY at the end of each timing plan cycle. Table B-2 is a listing of the data stored and the data fetched by this routine.

3.6.5 1.3 Assumptions

The tables to be copied are 20 locations in length.

3.6.5.1.4 Solution Approach

One table is updated for each call to the routine. The locations of the table to be copied and the table to be updated are passed from the calling routine to TCOPY through Accumulators AC2 and AC3.

3.6.5.1.5 Program Operation

When TNCY calls TCOPY, the beginning location of the table to be updated is in AC3. A counter, AC1, is initialized at -20. The contents of the address in AC2 are loaded into AC0 and stored at the address in AC3. AC1, AC2, and AC3 are then incremented and the load/store sequence is repeated. TCOPY loops back through the load/store/increment sequence until AC1 is zero and 20 elements have been copied. A return from subroutine TCOPY is then executed.

3.6.5.1.6 Model

Not applicable.

3.6.5.1.7 Date and Programmer

July 1975, N.G. Planer

3.6.5.1.8 Remarks

None.

3.6.5.2 Program Work Areas

3.6.5.2.1 Switches

None.

3.6.5.2.2 Counters, Accumulators

AC1 is used as a counter, referenced and altered by TCOPY.

3.6.5.2.3 Hold Areas, Buffers

The contents of the tables TPM1, TPM2, and TPM3, and TNTP are referenced by TCOPY. The contents of the tables TNTP and TCTP are altered by TCOPY.

3.6.5.3 Program Flowchart

The program flowchart is presented in Figure B-41.

3.6.5.4 Program Listing

The program listing is presented in Appendix D.

3.6.5.5 Input Data Formats

Not applicable.

3.6.5.6 Report Formats

Not applicable.

3.6.5.7 Test Data

To be determined.

3.6.5.8 Revisions

None.

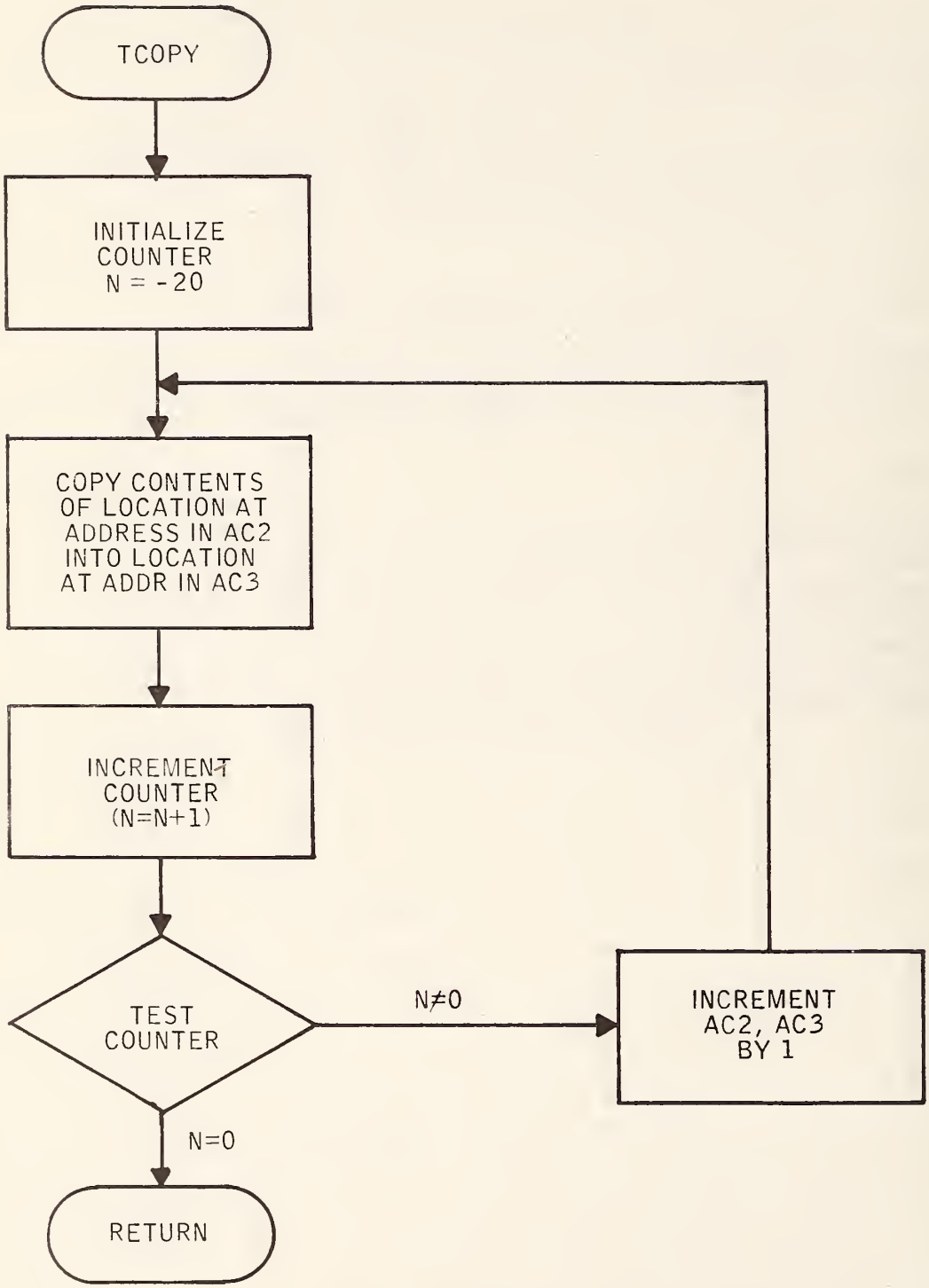


Figure B-41. Copy Timing Plan Tables Routine (Timing Plan Execution CPC)

3.6.6 Timing Plan Data Tables

"Timing Plan Data Tables" refers to the ROM storage area for fixed timing plan parameters. Definitions of these parameters follow.

TSCI - This is the index of the synchronization contractible interval. TSCI is not used in the current version of the PBD software; it may be used in later applications in which the timing plan CPC has more control of synchronization.

THI - This is the index of the transfer hold interval.

TAI1 - This is the index of the first adjustable interval in the timing plan.

TAI2 - This is the index of the second adjustable interval in the timing plan.

TIMX - This is the maximum value allowed for the interval index (TIN). If TIN exceeds TIMX during execution of the TNXT routine, an error is recorded, and a jump to the auto-disconnect routine (EADC) is executed.

TECI - This is the beginning location of a table containing contractible and extendible interval information. The table contains a parameter set for each of 16 possible detectorized intersection approaches. The locations are allocated as follows:

Relative Location	Contents
1	Primary contractible interval index
2	Secondary contractible interval index
3	Primary extendible interval index
4	Minimum contractible interval length

TCIW - This is the beginning location of the buffer for the command index word table. The buffer presently contains 20 locations, and for a timing plan with n intervals would be allocated as follows:

Relative Location	Contents
1	Command index word for first timing interval
2	Command index word for second timing interval
.	.
.	.
.	.
.	.
.	.
.	.
.	.
n	Command index word for nth timing interval

Any additional locations in the buffer are blank.

TPTAB - This is the beginning location of the three skeleton master timing plan tables. There are 60 locations, the last of which is labeled TPE. The first 20 locations contain the dial 1 skeleton master timing plan; the next 20 locations contain the dial 2 master; the third 20 locations contain the dial 3 master. All 60 locations are copied into the read/write data base tables at initialization, dial 1 into TPM1, dial 2 into TPM2, and dial 3 into TPM3.

TPE - This is the last location in the skeleton master timing plan tables. Its address is used as a signal to terminate the copy routine during initialization.

3.7 BUS PREEMPT CPC

This computer program component contains the functions for vehicle classification of the detector signature data, preempt modifications of the timing plan, and monitoring of preempt operations.

3.7.1 Bus Preempt Monitor (PMON)

3.7.1.1 Program Narrative

3.7.1.1.1 Purpose

This routine monitors preempt operation for initiating and terminating preempt modifications of the timing plan and determines the sequence in which the vehicle classifier examines the bus detector signature data base.

3.7.1.1.2 Software Interfaces

The preempt monitor routine (PMON) is called only by the executive scheduling routine, ESCH, during each real time clock interval. PMON calls VCLAS to examine a particular signature in the data base. The signature to be examined is specified by PMON through the detector index word in the data base, VCDT. The last datum in each signature is the end of signature status. Only completed signatures are classified. For classification of a signature as a bus, VCLAS returns to PMON with ACO set to zero; otherwise ACO is set to one. When a bus is classified, PMON calls the Preempt Request Routine, PREQ, for an upstream (preempt) detection and calls the Preempt Termination Routine, PTER, for only the downstream (clearance) detection on the preemptive approach.

3.7.1.1.3 Assumptions

There are two detectors on each preempt approach to the intersection: an upstream (preempt) detector and a downstream (clearance) detector. In the signature data base, the detectors are assigned to reserved blocks of signature data words. The two detectors on each approach are assigned to contiguous signature data blocks with the upstream detector preceding the downstream detector; thus, upstream detectors have odd indices and downstream detectors have even indices in the sequence.

When non-zero, the preempt in progress flag (PIP) has the value of the index number for the clearance detector on the preemptive approach to the intersection.

3.7.1.1.4 Solution Approach

A preempt in progress flag is used to indicate a preempt modification of the timing plan due to an upstream classification of a bus. A preempt in progress precludes any additional preempt modification, synchronization dwell, dial change, and programming panel update. At preempt initiation, a minimum duration timer, PMDT, is started. The preempt may not be terminated before this timer elapses. A preempt is terminated by a bus classification at the downstream detector on the same approach to the intersection as the upstream detector which detected the preempt vehicle. Preempt is also terminated when the preempt extension interval in the timing plan has elapsed.

The timing plan is structured to include both contractible intervals and the preempt extension intervals. Time is removed from contractible intervals and placed in the extending interval.

Extension time is always borrowed from other intervals so as to maintain the same synchronization status as prior to the preempt modification. The contracted intervals may be in the current cycle or the next cycle of the timing plan depending on the occurrence of the preempt initiation with respect to the timing plan.

The vehicle classification algorithm, VCLAS, is called to examine one specific signature in the data base. Only a completed signature is classified. The classification result (bus or not bus) is returned in a register to PMON.

For a preempt in progress, the clearance detector signature is tested first. Next, PMON cycles through all upstream detector signatures, followed by all downstream detector signatures.

A bus classification at any detector results in the starting of a timer for the purpose of the status display of classifier operation.

3.7.1.1.5 Program Operation

During each execution, PMON decrements the recent bus classification timer by one until zero is reached. Other routines initialize and test this timer. PMON also decrements the minimum preempt duration timer to zero. When the timer has elapsed, if the preempt in progress flag is non-zero, PMON calls the vehicle classifier routine, VCLAS, to examine the vehicle signature for the clearance detector that is indicated by the value of the flag. If the vehicle signature is classified as a bus, the Preempt Termination routine, PTER, is called to terminate the preempt condition. This is the only call to PTER. PMON then sets the detector index, VCDT, to one and VCLAS is called to examine the first upstream detector signature. Following the execution of VCLAS, VCDT is incremented by two (odd detector indices) and the operation is repeated until all upstream detector signatures have been examined. For each bus classification, the Preempt Request Routine is called. PMON then sets VCDT to 2 and calls VCLAS to examine the first downstream detector signature. Following the execution of VCLAS, VCDT is incremented by two (even detector indices) and the operation is repeated until all downstream detector signatures have been examined. Thus, all signatures are classified and recorded regardless of their affect on preempt operation. After processing the data for all detectors in the system, if a preempt situation still exists, PMON compares the extended interval index with the current interval index (TIN) to determine if the interval has expired. If it has expired, PMON clears the preempt in progress flag.

3.7.1.1.6 Model

Not applicable.

3.7.1.1.7 Date: July 11, 1975

Programmer: Robert Kyweriga

3.7.1.1.8 Remarks

None.

3.7.1.2 Program Work Areas

Refer to data base section for complete definitions

3.7.1.2.1 Switches

None.

3.7.1.2.2 Counters and Accumulators

PMDT - Preempt minimum duration timer

TIN - Current interval index

VCBT - Bus detected recently timer

Hold Areas

PIP - Preempt in progress flag

PMIF (3) - Preempt monitor index flags

VCDT - Vehicle detector index

Local data

PMHD - Set equal to DETS during program assembly, PMHD is the highest detector index number expected.

3.7.1.3 Program Flowchart

The program flowchart is presented in Figure B-42.

3.7.1.4 Program Listing

The program listing is presented in Appendix D.

3.7.1.5 Input/Output Data Formats

Not applicable.

3.7.1.6 Report Formats

Not applicable.

3.7.1.7 Test Data

To be determined.

3.7.1.8 Revisions

None.

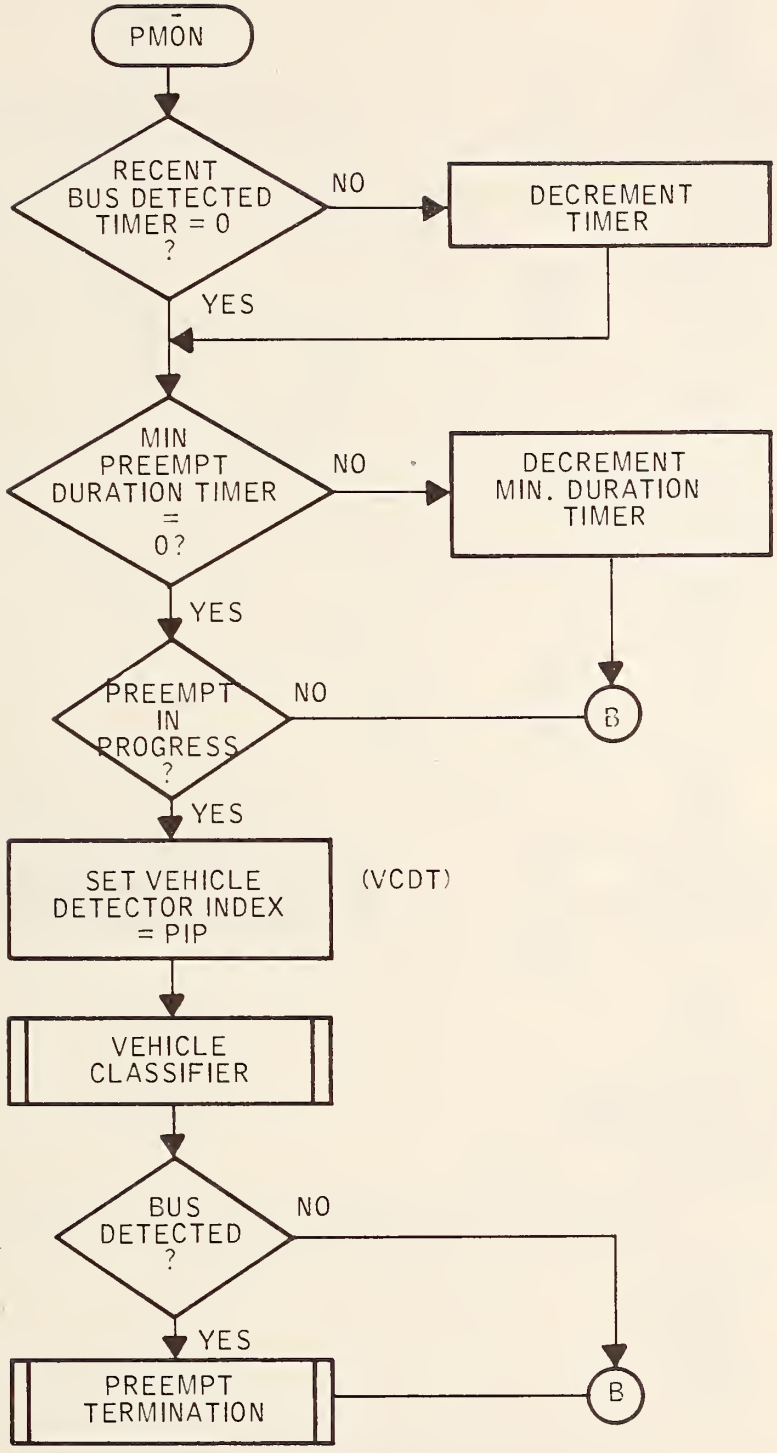


Figure B-42. Bus Preempt Monitor Routine (Bus Preempt CPC)

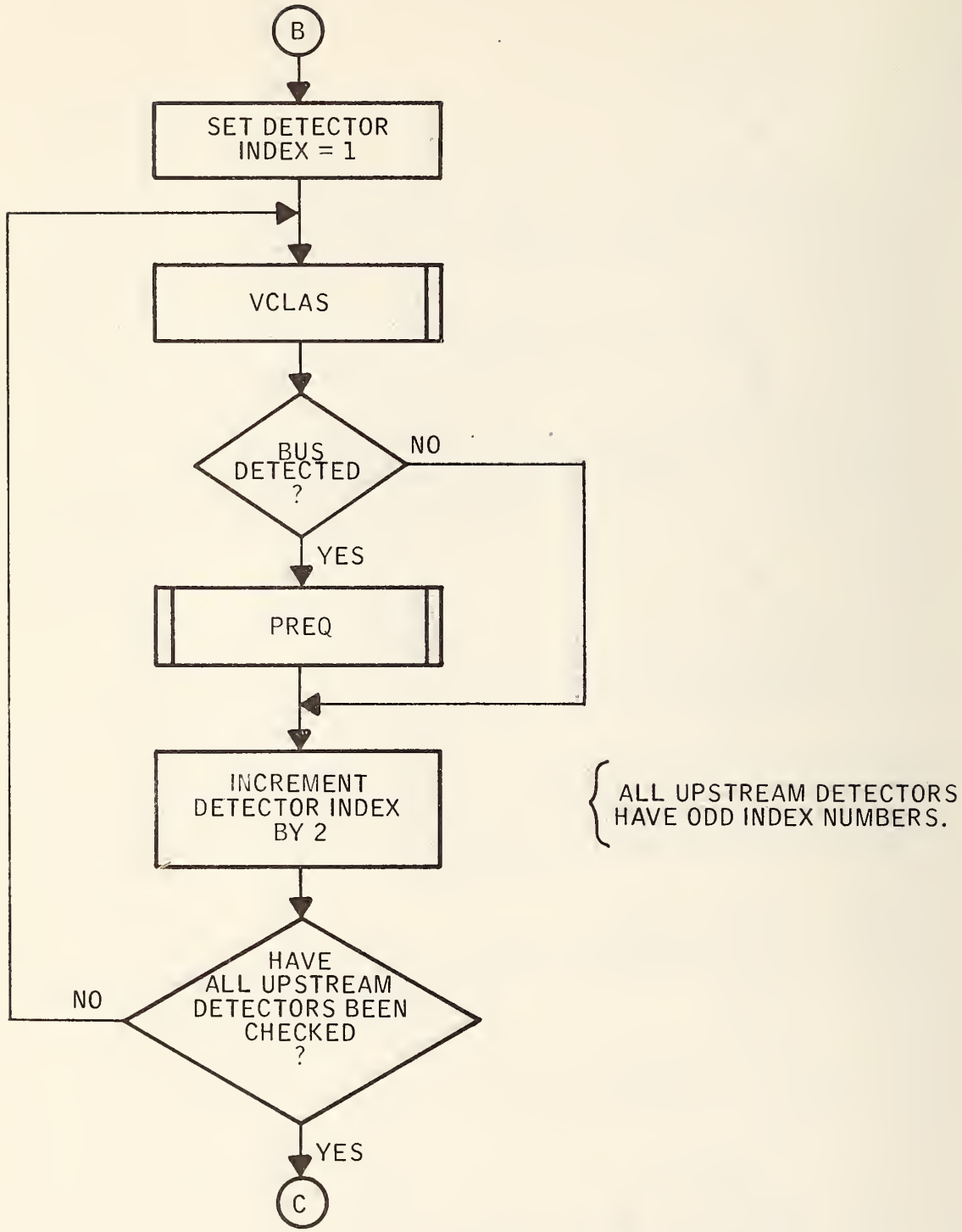


Figure B-42. Bus Preempt Monitor Routine (Bus Preempt CPC) (Continued)

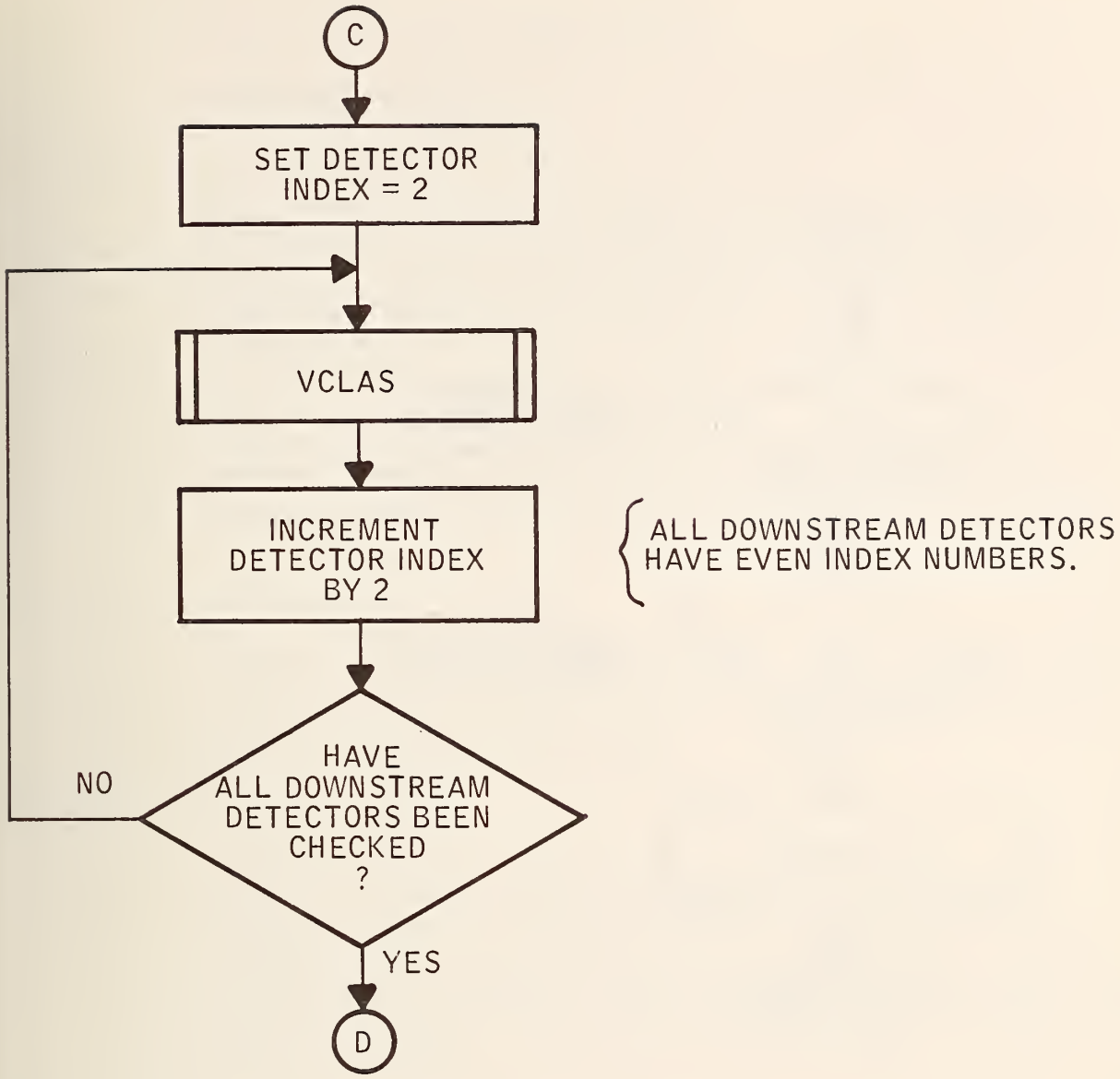


Figure B-42. Bus Preempt Monitor Routine (Bus Preempt CPC) (Continued)

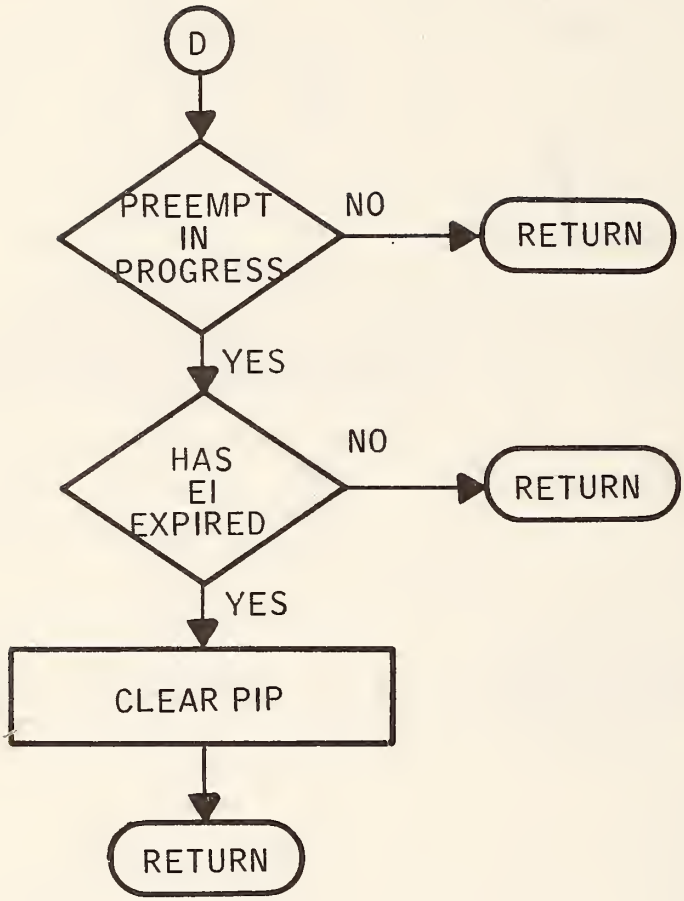


Figure B-42. Bus Preempt Monitor Routine (Bus Preempt CPC) (Concluded)

3.7.2 Preempt Request Routine (PREQ)

3.7.2.1 Program Narrative

3.7.2.1.1 Purpose

This routine initializes a preempt condition by setting the preempt in progress flag (PIP), determining which intervals should be modified, and modifying the duration of those intervals along with the duration of the current interval timer (TINT).

3.7.2.1.2 Software Interfaces

PREQ is called only by the preempt monitoring routine (PMON). PREQ uses the vehicle detector index number (VCDT) in determining which approach parameters should be used.

3.7.2.1.3 Assumptions

PREQ assumes that there is one set of approach parameters for every upstream detector and that these parameters are stored in blocks of four words arranged in consecutive order in the timing plan table, TECI.

3.7.2.1.4 Solution Approach

PREQ determines if preempt modifications are allowable. It then sets the preempt in progress flag, computes an address pointer to appropriate approach parameter block, and determines, on the basis of the current interval index (TINT), which intervals should be modified. Following these modifications to the durations of these intervals, PREQ returns to PMON.

3.7.2.1.5 Program Operations

PREQ begins by testing the WAIT/CONTROL flag. If in the WAIT state, PREQ returns PMON. The "bus preempt" switch on the programmable panel is tested and if off, PREQ returns to PMON. If the preempt in progress flag (PIP) is set, PREQ returns to PMON; if not, PIP is set with the index to the downstream clearance detector and an address pointer to the appropriate approach parameter block is generated and saved in AC3. The maximum extension for the current dial setting is copied into PEMAX. Indices to the primary contractible, secondary contractible, and extendible intervals

are determined, based on the current status of the timing plan. The modified intervals are in the current or the next cycle of the timing plan. The durations of the specified contractible intervals are reduced to the minimum duration specified by the approach parameters. The available time is used to increase the duration of the extendible interval. When the current interval index is the same as the primary contractible or extendible interval index, the current interval timer is modified to reflect the changes to the duration of the interval. The minimum duration timer, PMDT, is set to preclude a preempt termination before a reasonable traverse time from the upstream to the downstream detector.

The preempt request flag, PFREQ, is set to indicate a preempt modification of the timing plan and to preclude synchronization dwell. The wait for synchronization start flag, TEST, is reset to assure that the timing plan is no longer dwelling for synchronization.

3.7.2.1.6 Model

Not applicable.

3.7.2.1.7 Date: July 11, 1975 Programmer: Robert Kyweriga

3.7.2.1.8 Remarks

None.

3.7.2.2 Program Work Areas

Refer to data base section for complete definitions.

3.7.2.2.1 Switches

BPPD - Panel on/off preempt switch

3.7.2.2.2 Counters and Accumulators

PFREQ - Preempt request flag

PMDT - Preempt minimum duration timer

TIN - Current interval index

TINT - Current interval timer

Hold Areas

PEMAX - Maximum preempt extension for current dial setting

PERR - Preempt error buffer

PIP - Preempt in progress flag

PMIF (3) - Holds indices to modified intervals

PTCP - Address pointer to location of primary contractible interval in timing plan duration table

PTCS - Address pointer to location of secondary contractible interval in timing plan duration table

PTEI - Address pointer to location of extended interval in timing plan duration table

TCDS - Current dial setting

TCTP - Address pointer to current cycle timing plan table

TECI - Approach parameters

TFST - Wait for synchronized start flag

TNTP - Address pointer to next cycle timing plan table

VCDT - Detector index number

LOCAL DATA

PMAS - Masks used in determining settings of bus preempt panel switches

PRHD - Highest detector index number expected

PRLD - Lowest detector index number expected -1

PT18A - Address pointer to P18A

PT20 - Address pointer to P20

3.7.2.3 Program Flowchart

The program flowchart is presented in Figure B-43.

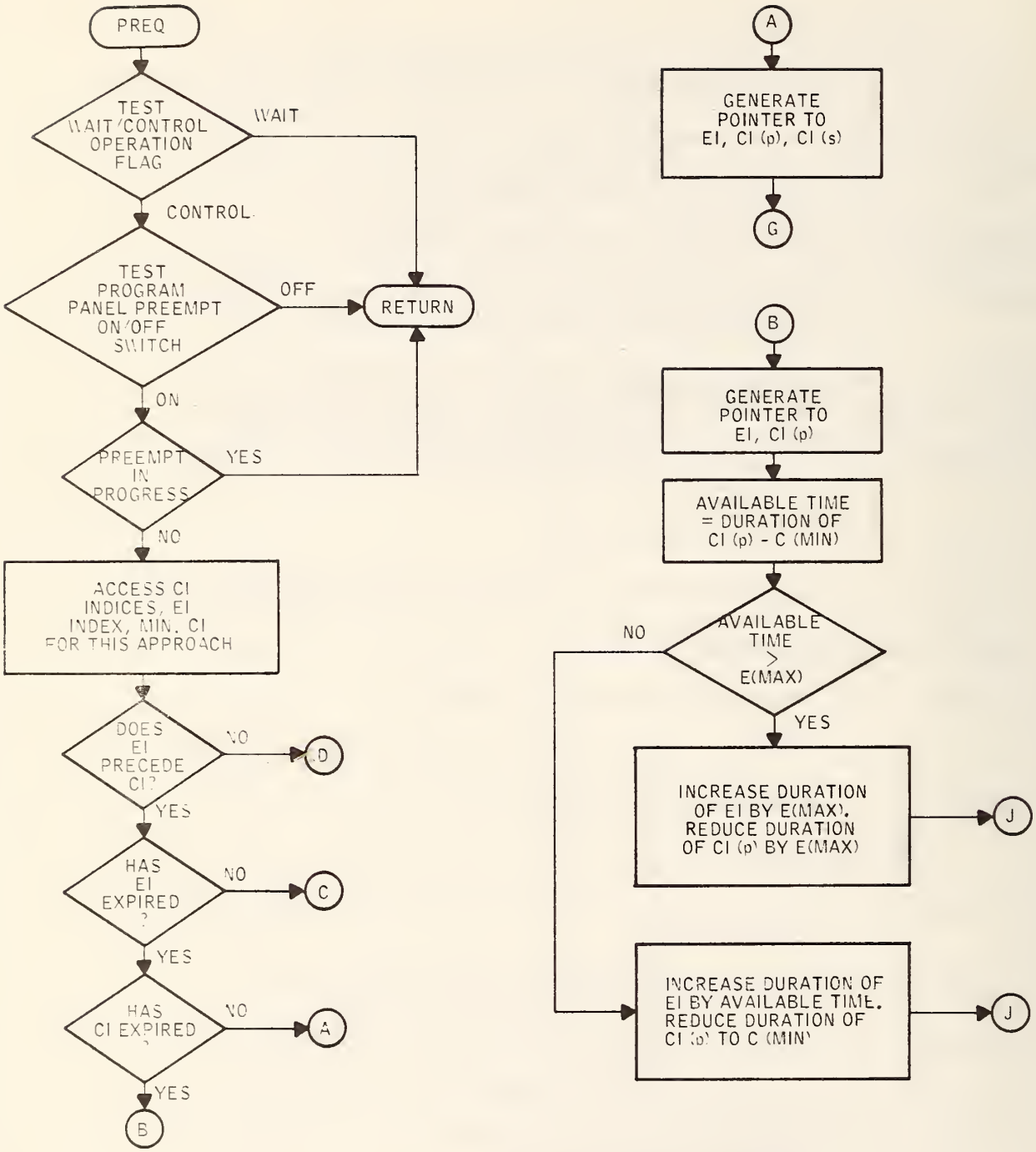


Figure B-43. Bus Preempt Request Routine (Bus Preempt CPC)

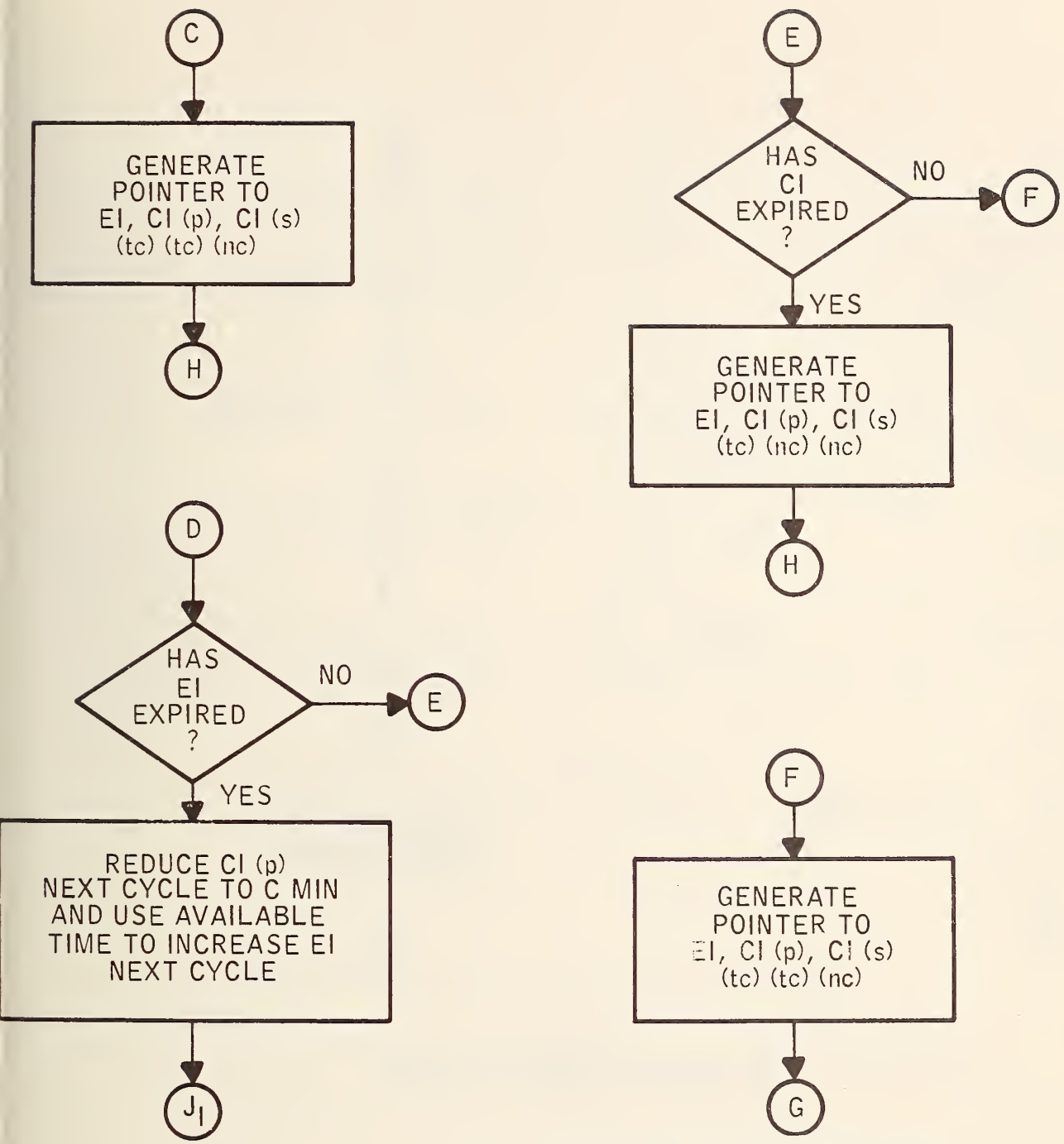


Figure B-43. Bus Preempt Request Routine (Bus Preempt CPC) (Continued)

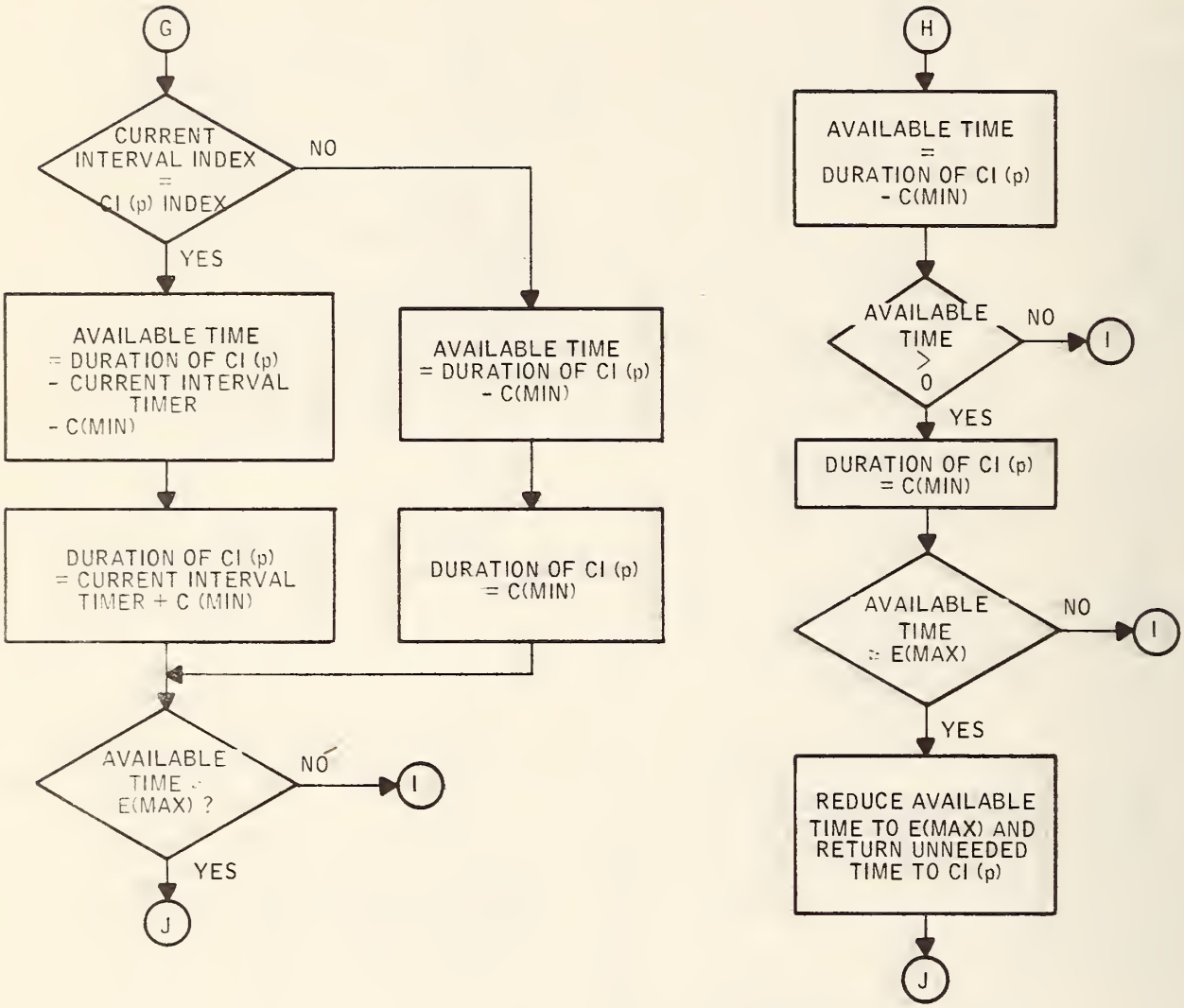


Figure B-43. Bus Preempt Request Routine (Bus Reempt CPC) (Continued)

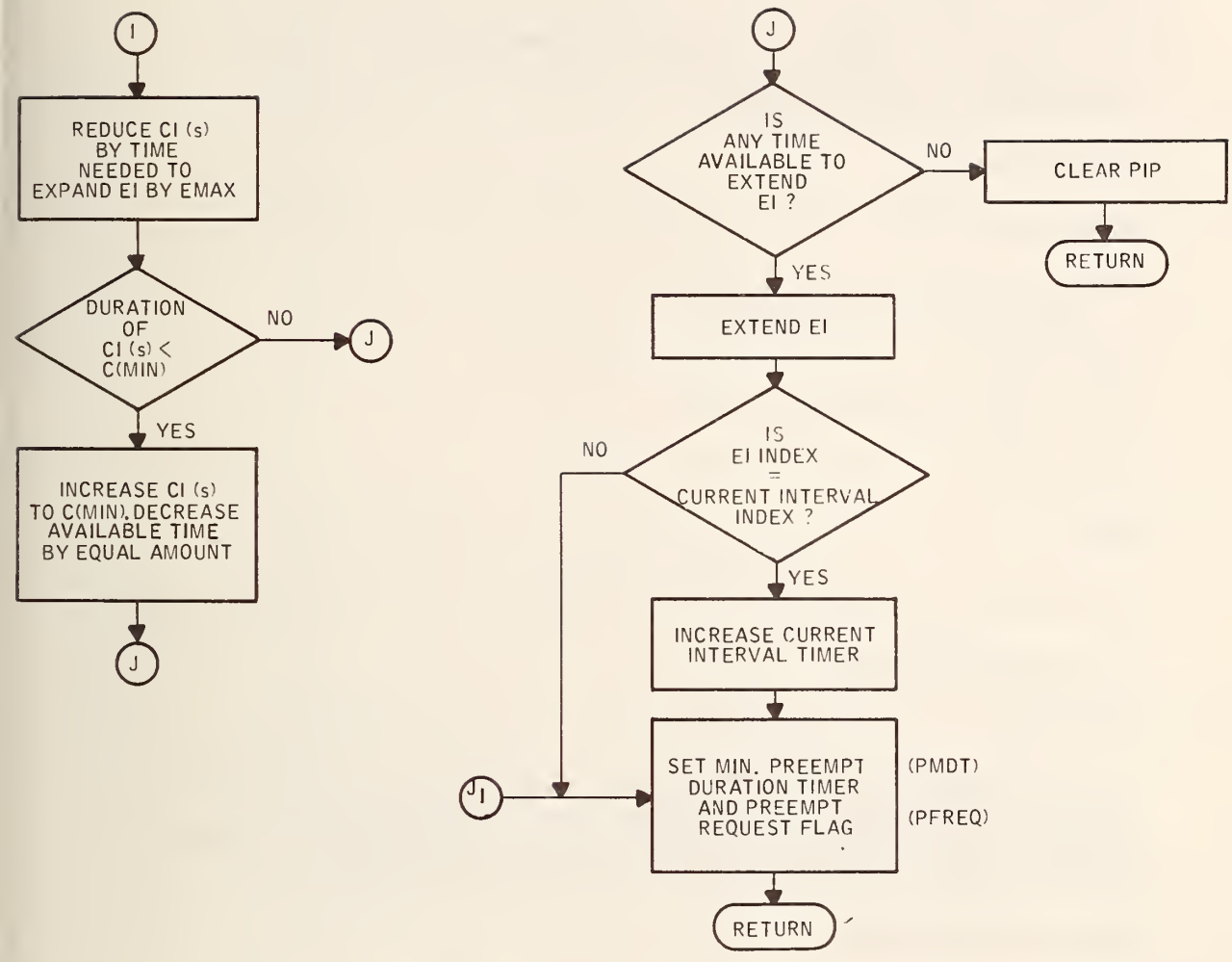


Figure B-43. Bus Preempt Request Routine (Bus Preempt CPC) (Concluded)

3.7.2.4 Program Listing

The program listing is presented in Appendix D.

3.7.2.5 Input/Output Data Formats

Not applicable.

3.7.2.6 Report Formats

Not applicable.

3.7.2.7 Test Data

To be determined.

3.7.2.8 Revisions

None.

3.7.3 Preempt Termination Routine (PTER)

3.7.3.1 Program Narrative

3.7.3.1.1 Purpose

This routine performs the preempt termination operations of clearing the preempt in progress flag, removing any unexpired preempt extension time from the extended interval, and returning such time to the contracted intervals.

3.7.3.1.2 Software Interfaces

The preempt termination routine is called only by the preempt monitor (PMON) at the classification of a bus at the clearance detector for the preempted approach. PTER calls PCIM to return unexpired preempt extension time to the contracted intervals. Information is transferred to PCIM using the variables, PTAVT, PTITE, and PTPMT. Refer to the data base section for definition of these variables.

3.7.3.1.3 Assumptions

Indices to the extended, primary contractible, and secondary contractible intervals have been stored in the PMIF array by the preempt request routine (PREQ).

3.7.3.1.4 Solution Approach

PTER begins by comparing the current interval index (TIN) with the index to the extended interval (EI). If the extended interval has expired, PTER clears the preempt in progress flag (PIP) and returns to PMON. If EI is a future interval, PTER reduces its duration to the value specified by the master timing plan duration table. If EI is the current interval, its duration is reduced to

$$\text{Duration} = \text{MAX}\ (D[EI_{(\text{master})}],\ D[EI] - TINT)$$

where $D[EI_{(\text{master})}]$ = Duration specified by master timing plan duration table

D[EI] = Duration specified by timing plan table

TINT = Current interval timer

The unexpired extension time is returned to the secondary and primary contractible intervals.

3.7.3.1.5 Program Operations

PTER begins by testing the WAIT/CONTROL flag for the CONTROL state. If in the WAIT state, PTER returns to PMON. The detector index number (VCDT) for the current bus classification is then tested against the preempt in progress flag (PIP). If they are not the same — if there is a clearance classification on other than the current preemptive approach, PTER returns to PMON. This test is redundant to the test in PMON. Address pointers are computed for the extended interval (EI) in the timing plan duration table, the first location of the master timing plan table, and EI in the master timing plan table. If the EI has expired, PTER returns to PMON. Otherwise, PTER reduces the duration of the EI to the value specified by the master timing plan. PCIM is then called to restore the unexpired extension time to the secondary contractible interval up to the duration in the master timing plan. Any additional extension time is then returned unconditionally to the the primary contractible interval. If the primary contractible interval has expired, the time is added to the corresponding primary contractible interval in the next cycle of the timing plan.

3.7.3.1.6 Model

Not applicable.

3.7.3.1.7 Date: July 11, 1975

Programmer: Robert Kyweriga

3.7.3.1.8 Remarks

None.

3.7.3.2 Program Work Areas

Refer to data base section for complete definitions

3.7.3.2.1 Switches

None.

3.7.3.2.2 Counters and Accumulators

TIN - Current interval index

TINT - Current interval timer

Hold Areas

PIP - Preempt in progress flag

PMIF (3) - Contains indices to intervals modified by PREQ

PTAVT - Available time for interval expansion

PTEM - Temporary storage word. Used in PTER to save address pointer to extended interval

PTITE - Index to interval to be expanded

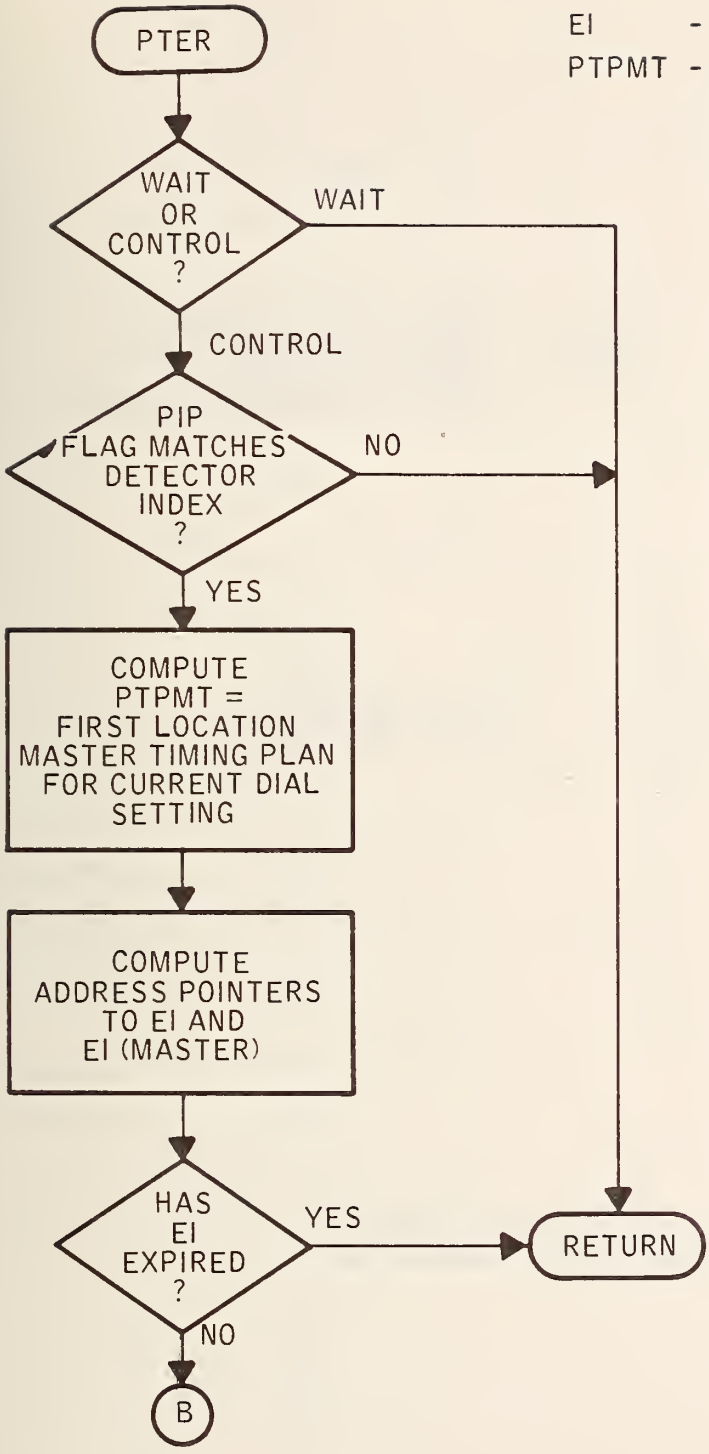
PTPMT - Pointer to current master duration timing plan.

3.7.3.3 Program Flowchart

The program flowchart is presented in Figure B-44.

3.7.3.4 Program Listing

The program listing is presented in Appendix D.



EI - EXTENDIBLE INTERVAL

PTPMT - PARAMETER USED IN CALLS TO PCIM. A POINTER TO THE FIRST LOCATION OF CURRENT MASTER TIMING PLAN.

Figure B-44. Bus Preempt Termination Routine (Bus Preempt CPC)

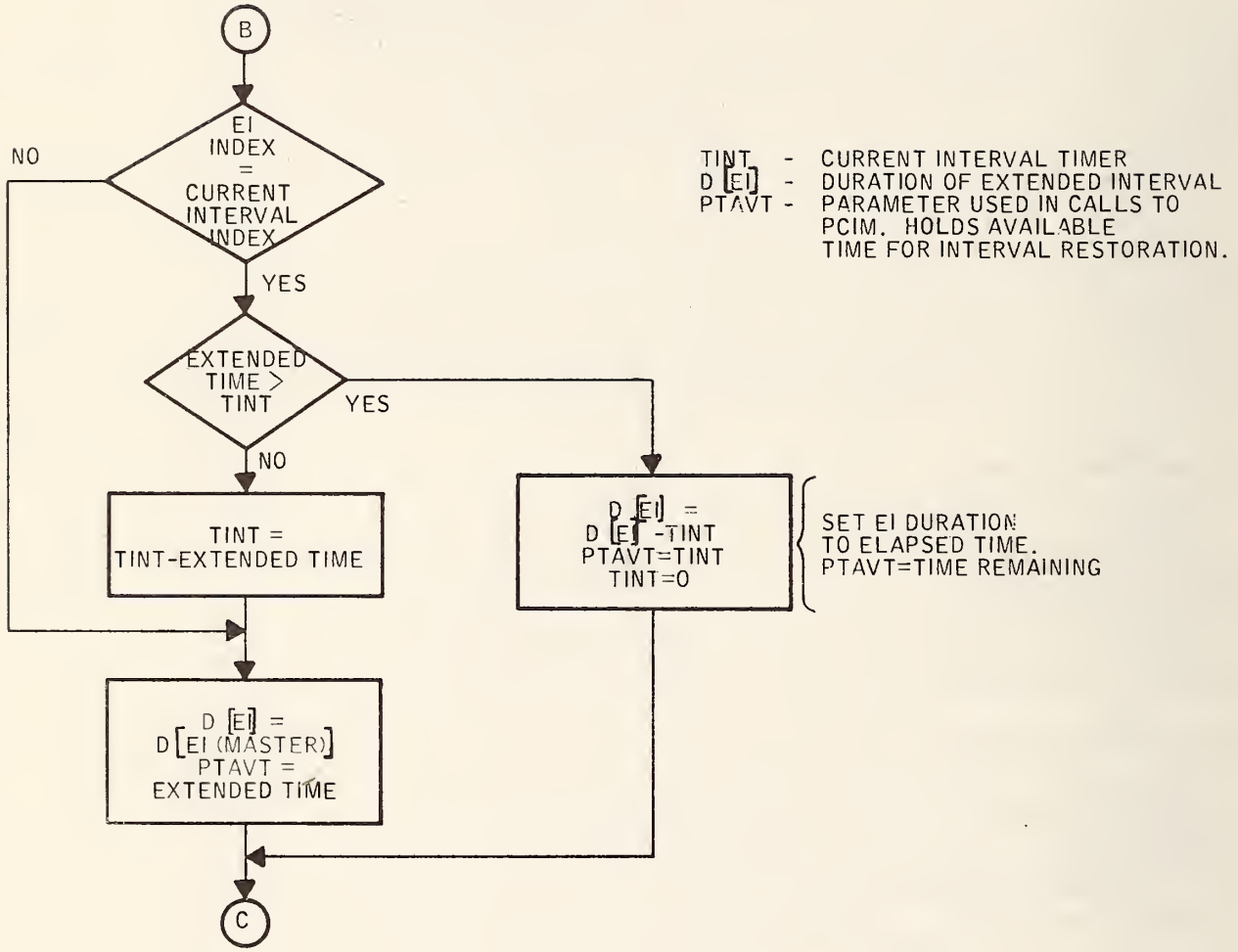


Figure B-44. Bus Preempt Termination Routine (Bus Preempt CPC) (Continued)

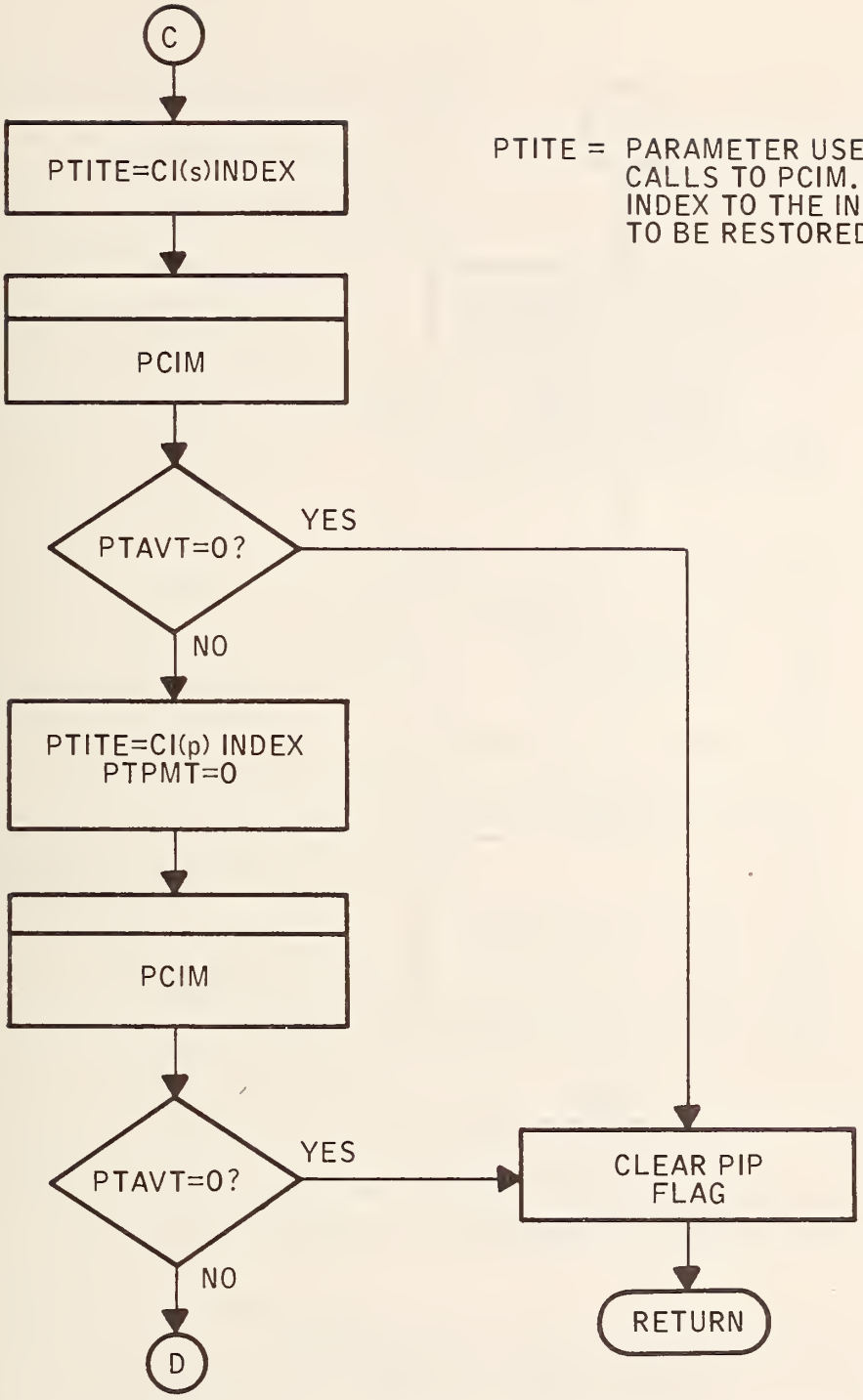


Figure B-44. Bus Preempt Termination Routine (Bus Preempt CPC) (Continued)

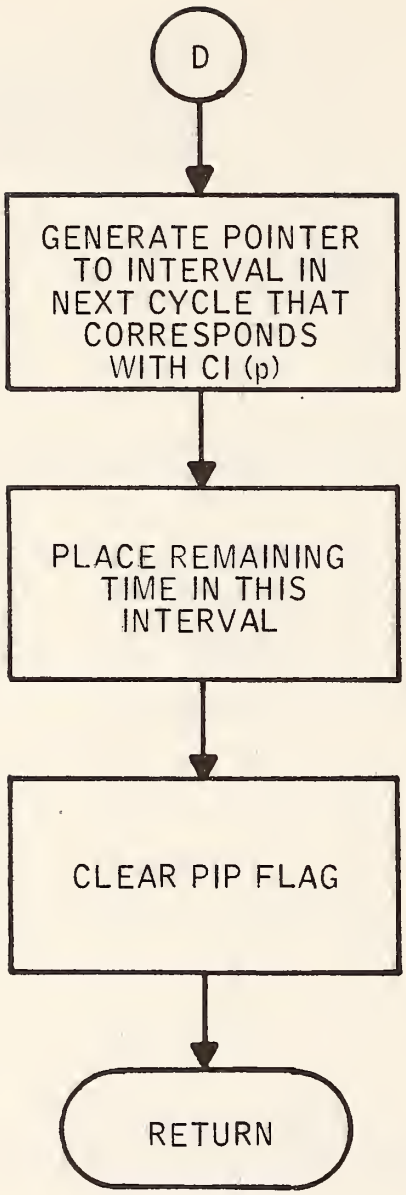


Figure B-44. Bus Preempt Termination Routine (Bus Preempt CPC) (Concluded)

3.7.3.5 Output Data Formats

Not applicable.

3.7.3.6 Report Formats

Not applicable.

3.7.3.7 Test Data

To be determined.

3.7.3.8 Revisions

None.

3.7.4 Vehicle Classifier

3.7.4.1 Program Narrative

3.7.4.1.1 Purpose

This routine classifies vehicle signatures as that of a standard 40-foot transit bus or not as a bus.

3.7.4.1.2 Software Interfaces

The vehicle classifier routine (VCLAS) is called by preempt monitor (PMON) to examine the data placed in the SDB by a specific detector. The detector block to be examined is specified by PMON through a word in the data base (VCDT). If the EOS status word for this block equals one, the block is copied from the SDB to the first seven words of the vehicle classifier data base (VCDB) and is classified as bus or non-bus. If vehicle is a bus, ACO is set to zero; otherwise, ACO is set to one. VCLAS then returns to PMON. Errors detected are recorded in the preempt error record buffer (PERR) by the proper error handling routine.

3.7.4.1.3 Assumptions

The vehicle classifier is concerned with distinguishing between "standard transit" buses and other vehicles.

3.7.4.1.4 Solution Approach

VCLAS is given a pointer index (VCDT) to a segment of the SDB that contains the vehicle signature data items gathered

by a specific detector. VCLAS decides whether the data items meet the required criteria, whether the vehicle should be evaluated by the moving or stopped vehicle classifier, and whether the vehicle is a bus or non-bus. If an error is detected, the incident is recorded in PERR by the proper error handling routine. VCLAS will inform PMON whether the vehicle was a bus or non-bus. VCLAS does not call routines to modify the timing plan.

3.7.4.1.5 Program Operation

When called by PMON, VCLAS begins by examining the EOS status word for the specified detector. If the EOS status word is zero, ACO is set to one and the program returns to PMON; otherwise, the vehicle signature data items are copied to VCDB. Following the copy operation, the EOS status word is checked again to determine if it has changed. At the start of a new vehicle signature, the EOS status word and the third peak amplitude value are set to zero by the preprocessor unit. If the status word has not changed or if it has changed but the third peak amplitude value is greater than zero, VCLAS continues. Otherwise ACO is set to one and the program returns to PMON. Next, the data items are examined to insure that range, data order, and gross feature criteria are met. If these restrictions are not met, the errors detected are recorded, ACO is set to one, and the program returns to PMON. If the restrictions are satisfied, an initial evaluation determines whether the "stopped" or "moving" vehicle classification algorithm should be employed. The chosen algorithm is then applied, the vehicle is classified as bus (ACO set to zero) or non-bus (ACO set to one), and the program returns to PMON.

3.7.4.1.6 Model

Not applicable.

3.7.4.1.7 Date: June 4, 1975 Programmer: Robert Kyweriga

3.7.4.1.8 Remarks

None.

3.7.4.2 Program Work Areas

3.7.4.2.1 Switches

None.

3.7.4.2.2 Counters and Accumulators

VCVC(32) - Vehicle classifier vehicle counter array with one word reserved for each detectors. Used for detector failure monitoring. Value stored within a word is incremented when the corresponding detector has placed a valid vehicle signature in SDB.

VCLOG(12) - First word of the Vehicle Classifier Vehicle log. Statistics such as number of buses, non-buses, moving vehicles, and stopped vehicles are recorded here. Located in data base, these counters are for all vehicles, all detectors, and are limited to less than 2^{32}.

VCLOG and VCLOG + 1 - A double precision word containing the number of signatures rejected as noise.

VCLOG + 2 and VCLOG + 3 - A double precision word containing the number of signatures that failed the gross features test.

VCLOG + 4 and VCLOG + 5 - A double precision word containing the number of signatures that were handled by the stopped vehicle classifier.

VCLOG + 6 and VCLOG - 7 - A double precision word containing the number of signatures that were handled by the moving vehicle classifier.

VCLOG + 8 and VCLOG + 9 - A double precision word containing the number of signatures classified as buses.

VCLOG + 10 and VCLOG + 11 - A double precision word containing the number of signatures classifed as non-buses.

PERR(17) - Preempt Error Buffer

PERR(1) - Error Flag

PERR(2) - Number of overflow errors detected

PERR(3) - Specified detector during last overflow error

PERR(4) - Number of times specified detector was out of range

PERR(5) - Specified detector during last such error

PERR(6) - Number of times an amplitude value was out of range

PERR(7) - Specified detector during last such error

PERR(8) - Number of times a time value was out of range

PERR(9) - Specified detector during last such error

PERR(10) - Number of times a time value was out of sequence

PERR(11) - Specified detector during last such error

PERR(12) - Number of vehicle signatures rejected as noise

PERR(13) - Specified detector during last such error

PERR(14) - Number of times EOS status word changed during copy operation

PERR(15) - Specified detector during last such error

PERR(16) - Number of times a signature was lost during a copy operation

PERR(17) - Specified detector during last such error

Hold Areas

VCDB(16) - Vehicle Classifier Data Base. This array in the data base contains information used and produced by the vehicle classifier routine.

VCDT - Detector index number specifies which detector data block is to be examined VCLAS. Located in data base.

SDB(32,8) - Signature data block contains all detector data blocks.

Detector Index Range

VCHD - Highest detector index number expected

VCLD - Lowest detector index number -1

Masks

VCMA - Mask used in amplitude in-range test

VCMT - Mask used in time in-range test

VCMOF - Mask used in time overflow test

Constants used in moving/stopped test

VCST1 - Evaluation term for F1 + F2

VCST2 - Evaluation term for F0 (lower bound for moving vehicle)

VCST3 - Evaluation term for F0 (upper bound for moving vehicle)

Constants used in stopped vehicle classifier

VC31 - G1 coefficient for H1 function

VC32 - G2 coefficient for H1 function

VC41 - G1 coefficient for H2 function

VC42 - G2 coefficient for H2 function

H1T1 - Evaluation term for H1 function (lower bound for bus)

H1T2 - Evaluation term for H1 function (upper bound for bus)

H2T1 - Evaluation term for H2 function (lower bound for bus)

H2T2 - Evaluation term for H2 function (upper bound for bus)

VCM15 - Mask used for double precision comparisons

VCCO - Term for clearing overflow status bit

VCCO+1 - F1 coefficient for K4

VCCO+2 - F2 coefficient for K4

VCCO+3 - G1 coefficient for K4

VCCO+5 - K4 Bias (residue)

VCCO+6 - F1 coefficient for K5

VCCO+7 - F2 coefficient for K5

VCCO+8 - G1 coefficient for K5

VCCO+9 - G2 coefficient for K5

VCCO+10 - K5 Bias (residue)

3.7.4.3 Program Flowchart

The program flowchart is presented in Figure B-45.

3.7.4.4 Program Listing

The program listing is presented in Appendix D.

3.7.4.5 Input Data Formats

Not applicable.

3.7.4.6 Report Formats

Not applicable.

3.7.4.7 Test Data

To be determined.

3.7.4.8 Revisions

None.

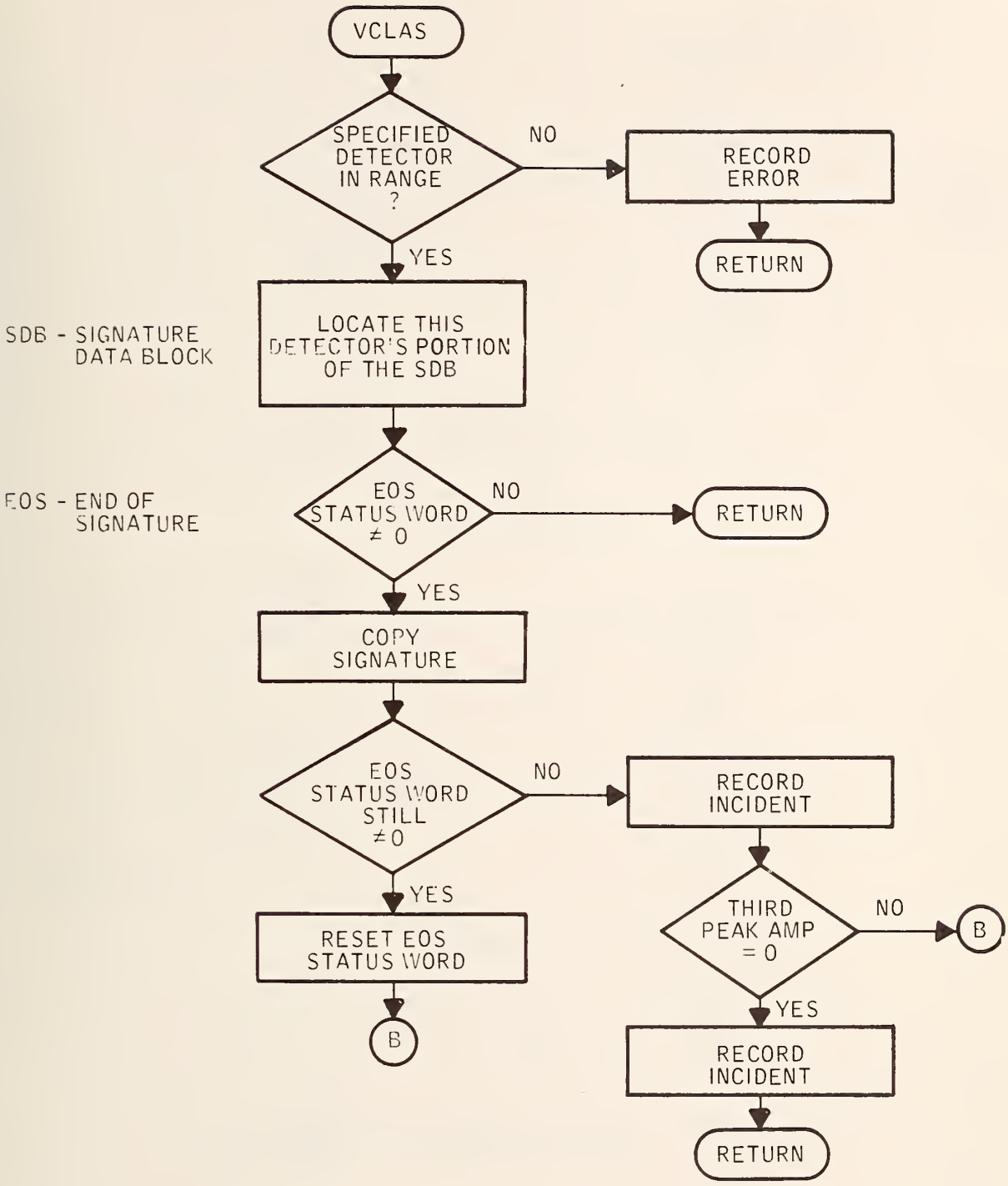


Figure B-45. Vehicle Classification Routine (Bus Preempt CPC)

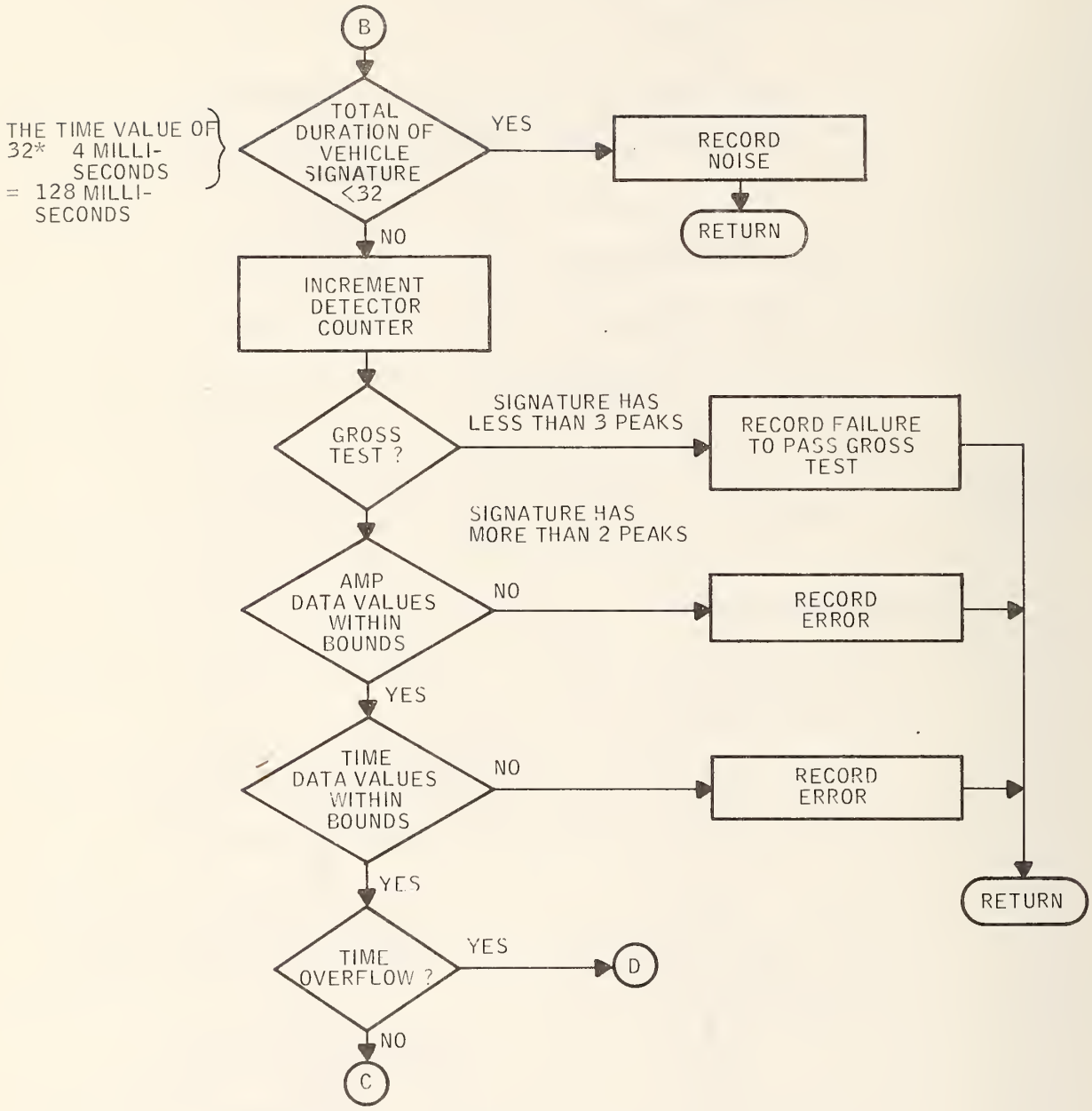


Figure B-45. Vehicle Classification Routine (Bus Preempt CPC) (Continued)

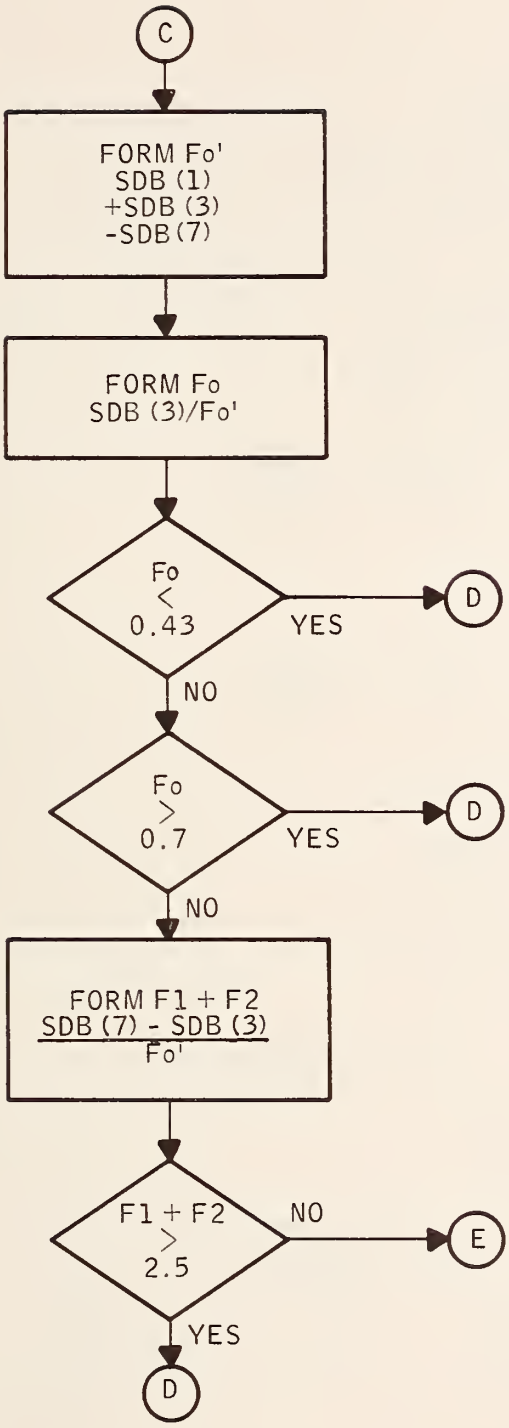


Figure B-45. Vehicle Classification Routine (Bus Preempt CPC) (Continued)

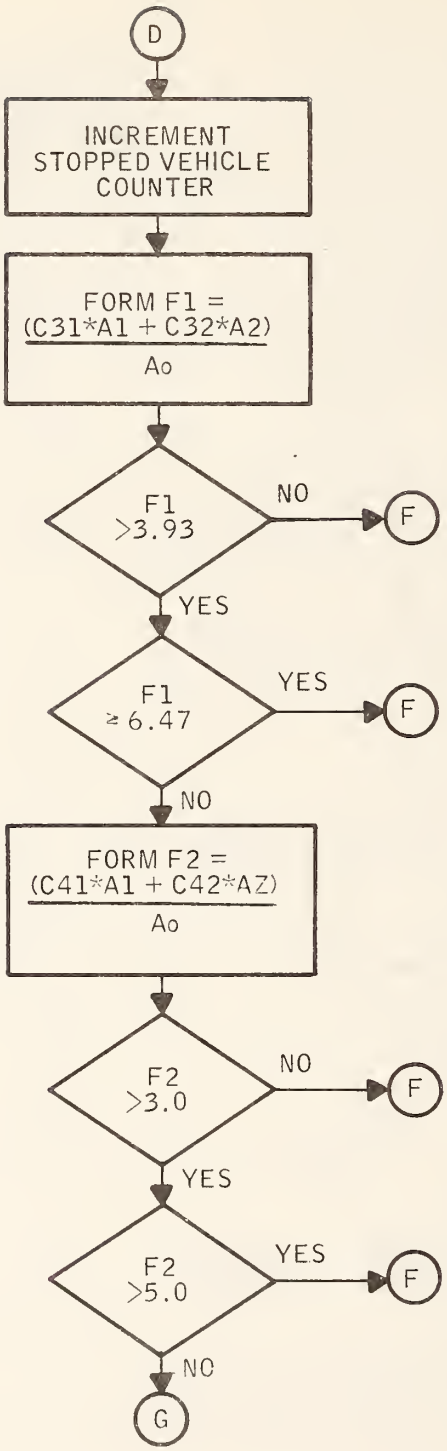


Figure B-45. Vehicle Classification Routine (Bus Preempt CPC) (Continued)

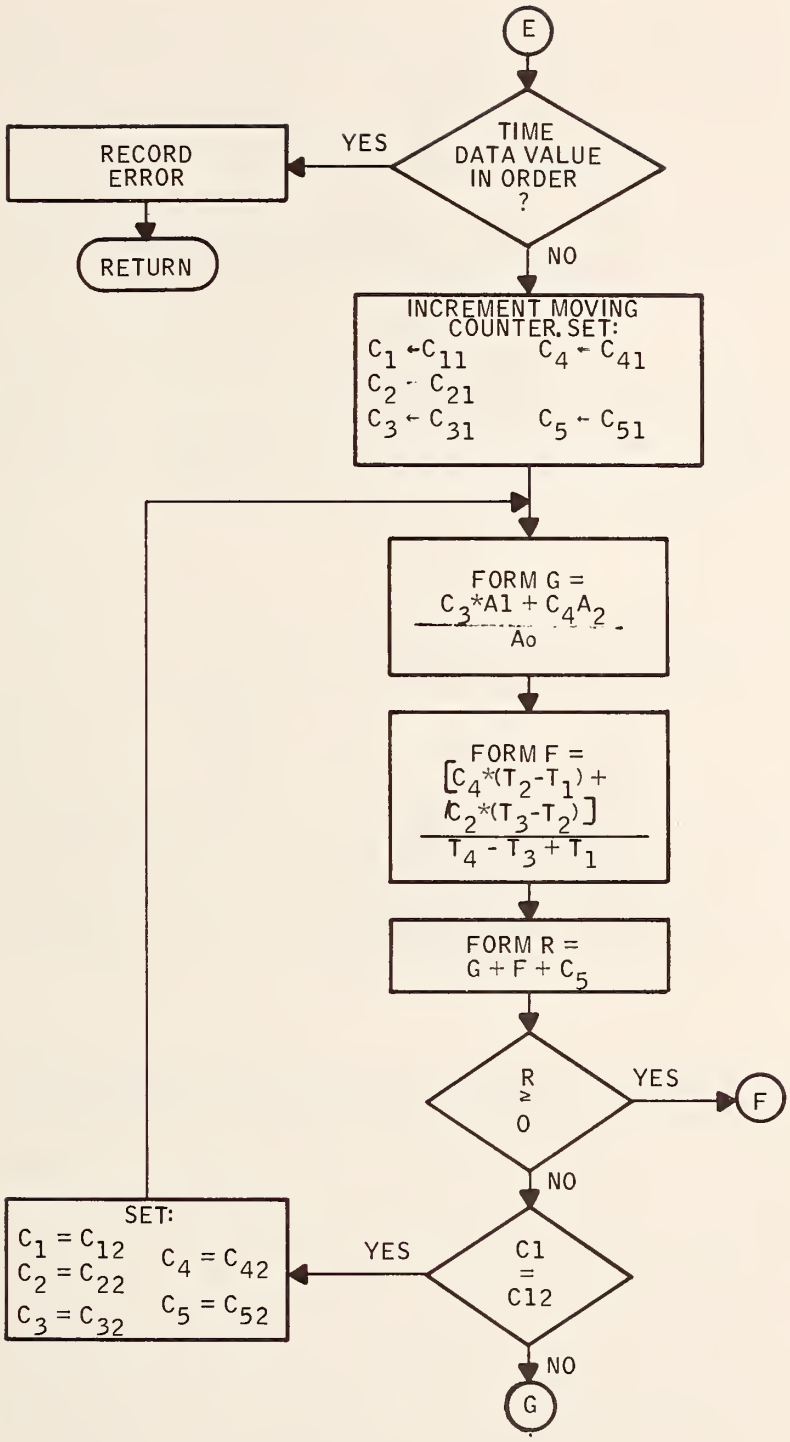


Figure B-45. Vehicle Classification Routine (Bus Preempt CPC) (Continued)

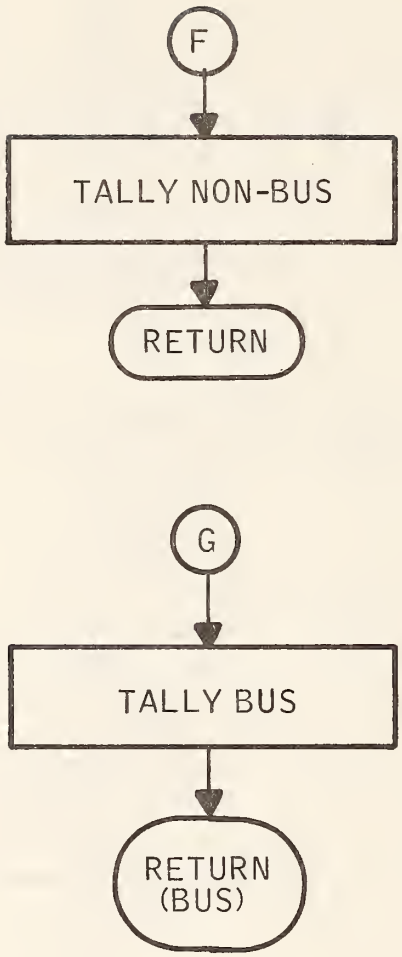


Figure B-45. Vehicle Classification Routine (Bus Preempt CPC) (Concluded)

3.7.5 Preempt Interval Modifier (PCIM)

3.7.5.1 Program Narrative

3.7.5.1.1 Purpose

This routine returns unexpired preempt extension time to the contracted intervals.

3.7.5.1.2 Software Interfaces

PCIM is called only by the preempt termination routine (PTER). Information is transferred to PCIM using the variables, PTAVT, PTITE, and PTPMT. Refer to the data base section for definition of these variables.

3.7.5.1.3 Assumptions

PTPMT when non-zero contains the first location of the master timing plan for the current dial setting.

3.7.5.1.4 Solution Approach

PCIM first generates an address pointer to the specified contracted interval (PTITE). If PTPMT is zero, the interval is extended by the time available (PTAVT); if PTPMT is the first location of the master timing plan for the current dial setting, expansion of the interval is limited to the duration specified in the master timing plan duration table.

3.7.5.1.5 Program Operation

PCIM begins by generating an address pointer to the contracted interval (CI) from the interval index provided in PTITE. PCIM then determines if CI has expired, and if so, returns to PTER. If PTPMT is non-zero, it contains an address pointer to the first location of the master timing plan table for the current dial setting. This is used to generate a pointer to the interval in the master timing plan (CI master) that corresponds with CI. Expansion of CI is then limited to the value specified in CI master. If PTPMT is zero, expansion of CI is unlimited and all available time is used to increase the duration of CI. If the current interval index (TIN) does not equal the index to CI (PTITE), PCIM returns to PTER; otherwise, PCIM adjusts the current interval timer (TINT) and then returns to PTER.

3.7.5.1.6 Model

Not applicable.

3.7.5.1.7 Date: July 11, 1975

Programmer: Robert Kyweriga

3.7.5.1.8 Remarks

None.

3.7.5.2 Program Work Area

Refer to data base section for complete definitions.

3.7.5.2.1 Switches

None.

Hold Areas

ATCTP - Address pointer to the current cycle timing plan table

ATNTP - Address pointer to the next cycle timing plan table

PTAVT - The amount of time available to expand the specified interval

PTEM - Temporary storage location

PTITE - The index to the interval to be expanded

PTPMT - When non-zero, the first location of the master timing plan for the current dial setting

TIN - The current interval index

TINT - The current interval timer

3.7.5.3 Program Flowchart

The program flowchart is presented in Figure B-46.

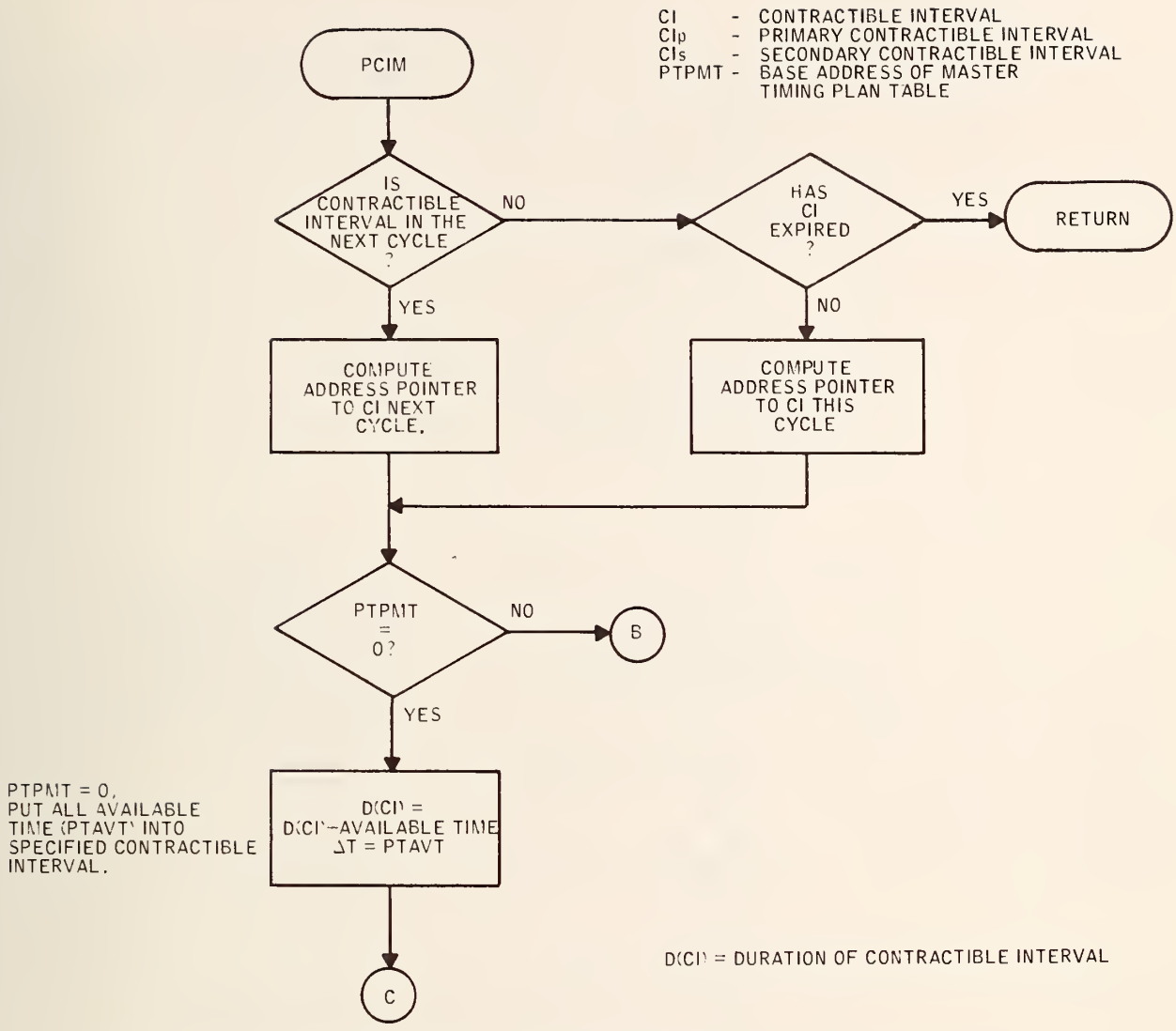


Figure B-46. Preempt Interval Modifier (Bus Preempt CPC)

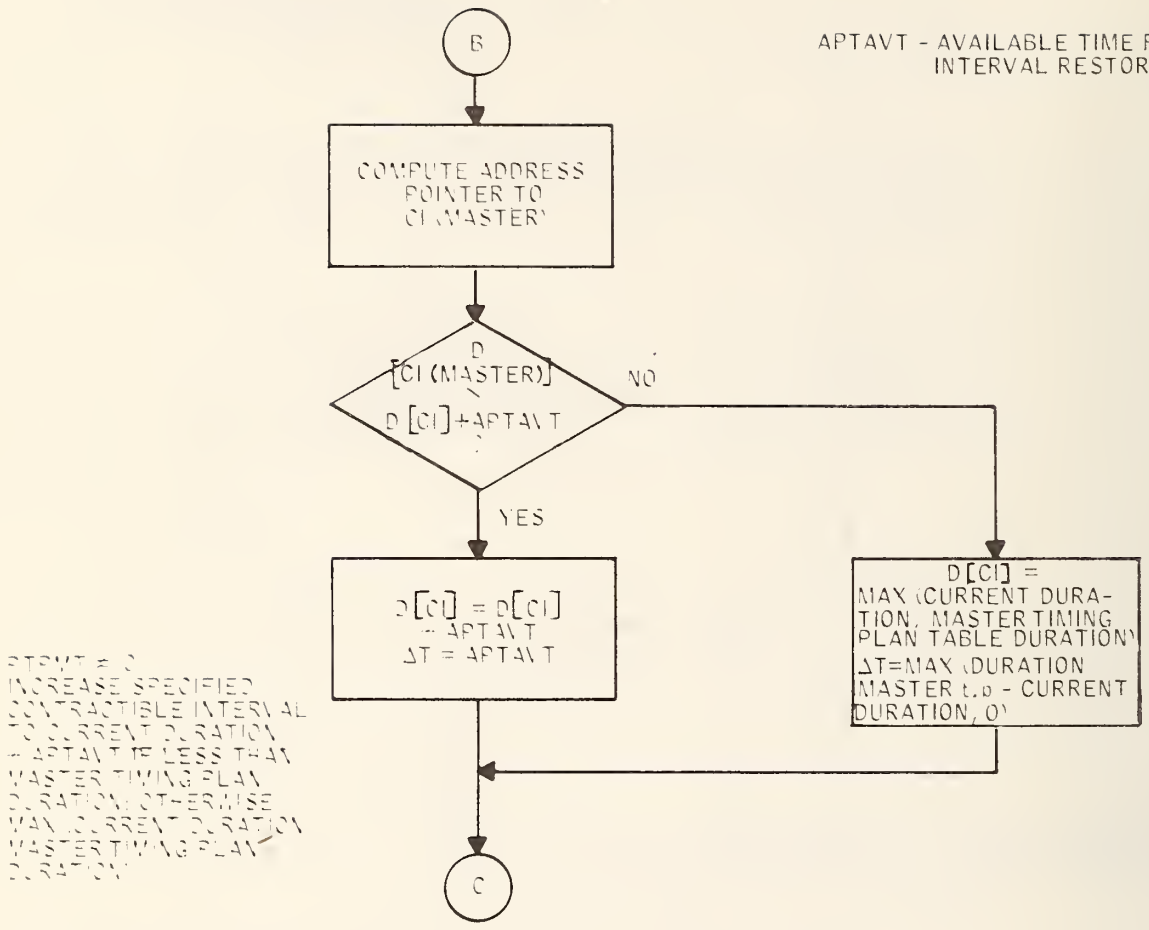


Figure B-46. Preempt Interval Modifier (Bus Preempt CPC) (Continued)

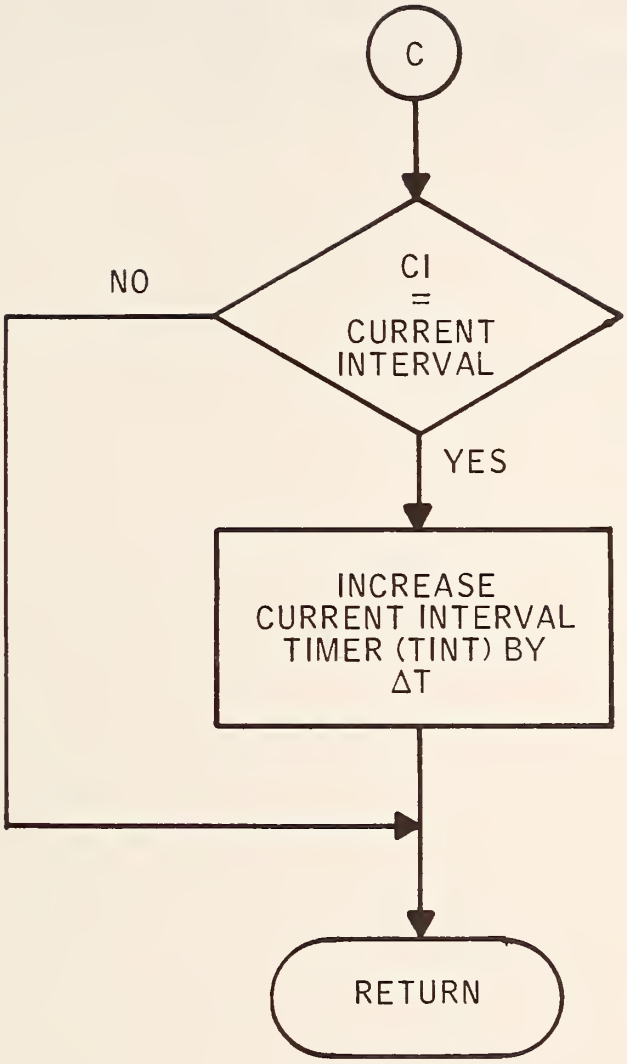


Figure B-46. Preempt Interval Modifier (Bus Preempt CPC) (Concluded)

3.7.5.4 Program Listing

The program listing is presented in Appendix D.

3.7.5.5 Input/Output Data Formats

Not applicable.

3.7.5.6 Report Formats

Not applicable.

3.7.5.7 Test Data

To be determined.

3.7.5.8 Revisions

None.

3.8 PROGRAM LISTINGS AND FLOWCHARTS

The complete program listing including the assembly constants and the cross reference table of programming symbols is presented in Appendix D. The flowcharts are included with the program documentation of each routine.

Table B-1. Memory Size of Assembled Routines

Routine	Size (Hexadecimal Words)	Address Range (Hexadecimal)	Spare (Hexadecimal Words)
ERSI EGIS ECER ECCU EMAD ESCH EADC COTF	41 7E 3 1F 13 1C 33 47	1000 - 11FF	76
EDIS ECOM ESYN EMIC ETIM EDET ESDW EEOC EREO-ERE12	12 2D 43 21 F 2E 2B 25 75	1200 - 13FF	4B
EPAN EPPM EPP1 EPP2 EPP3 EPUP EPBI EM10 EM24	6E 56 B5 B4 B4 46 27 5 5	1400 - 17FF	49
TPEN TSST TNNT TNCY TCOPY Timing Plan Tables	18 19 35 63 A 95*	1800 - 19FF	98
PMON PCIM PTER PREQ VCLAS	35* 3A* 5E* 133* 18E	1A00 - 1DFF	59
EBKD SMAD	C 15	1E00 - 1EFF	DF
BRTC BPAN NRTC NSDC NLON1 NLOF1 NLON2 NLOF2	D 1B E 61 3 3 3 3	1F00 - 1FFF	50

* These modules are application-dependent.

Table B-2. RAM Data Store/Fetch Matrix

SORTED BY ROUTINE

Routine	Data Values Stored	Data Values Fetched
ERSI	All locations from VCSDB to DEND), ERAM, ERTCF, ETCY+2, TPM1, TPM2, TPM3, TIN, TINT, TCIWC	
EGIS	ESAV, ISV, ERTCF, EERR	ERTCF, EERR, ESAV
EBKD	SSAD, SEAD	
ECER		
ESCH		
EADC	EDSAV, XSDW	XSDW
EDIS		EDIN
ECOM	ECMF, EERR	XOCW1, XOCW2, BCR1, BCR2, ECMF, EERR
ESYN	EERR, ESYD, EPST	ER123, ESYD, TCDS, EERR
EMIC	EDNO, ERNO, ED23, ER123	BDIN, EDNO, ERNO
ETIM	ETCY, ETST, EDMIT	BDIN, ETCY, EDMIT
EDET	EDETT, EERR, EDERR, VCVC, EDMIT	EDMIT, VCVC, EDETT, EERR
ESDW	XSDW	ER123, ED23, EPST, PIP, VCBT, EERRF, XERRF, BERRF, TERRF, PERRF, BDIN
EEOC	TFST, PFREQ, PMIF, ETCY, ETCY+1, ETCY+2	PEREQ, PMIF, ETCY, ETCY+1, ETCY+2
EREO-ERE12	PERR, EERR, EPEF1, EPEF2, EPEF3	PERR, EERR, EPEF1, EPEF2, EPEF3
EPAN	EP1E, EP2E, EP3E, EPDVE, EPFT	EP1E, EP2E, EP3E, EPDVE, TPM1, TPM2, TPM3
EPPM	EPUF, EPIR, EPCF, EP1E, EP2E, EP3E, EPDVE	BPPD, EP1R, EPCF, EP1E, EP2E, EP3E, EPDVE
EPP1	EPD1, EPCD1, EP1E, EPDVE	BPD1, EPFT, EPD1
EPP2	EPD2, EPCD2, EP2E, EPDVE	BPD2, EPFT, EPD2
EPP3	EPD3, EPCD3, EP3E, EPDVE	BPD3, EPFT, EPD3
EPUP	TPM1, TPM2, TPM3, TDWL, PEMAX, ESYD	EPCD2, EPCD3
EPBI (EPBI3, EPBI2, EPBI1)		
EM10, EM24		
TPEX	TCIWC, TINT	TIN, BDIN, TINT
TSST	TFST	TFST, TCDS, TDWL, ETST, EPST
TNXT	TIN, TINT, TCIWC, TERR	TIN, TCTP, EDIN, TERR

Table B-2. RAM Data Store/Fetch Matrix (Continued)

SORTED BY ROUTINE (Continued)

Routine	Data Values Stored	Data Values Fetched
TNCY	CTST, TFST, EPUF, TERR, TCDS	PIP, BPPD, TCDS, EPUF, ED23, TERR
TCOPY	TCTP, TNTP	TNTP, TPM1, TPM2, TPM3
PMON	VCBT, PMDT, VCDT, PMIF, PIP	VCBT, PMDT, PIP, VCDT, PMIF, TIN
PCIM	PTEM, PTITE, PTAVT, TINT	PTAVT, PTITE, TIN, TINT, TCTP, TNTP, PTPMT, PTEM
PTER	CTST, PTEM, PTPMT, TINT, PTAVT, PTITE, TCTP, TNTP, PIP	ETCY, BDIN, PIP, VCDT, PMIF, TCTP, TNTP, TCDS, PTEM, TIN, TINT
PREQ	CTST, PIP, PEMAX, PMIF, PTCS, PTEI, PTCP, TCTP, TNTP, TINT, PMDT, PEREQ, TFST	ETCY, BDIN, TCDS, BPPD, VCDT, PIP, PEMAX, TIN, TINT, TNTP, TCTP, PMIF, PTCS, PTEI, PTCP
VCLAS	VCDB, VCLOG, VCVC, VCBT	VCDT, VCSDB, VCLOG, VCVC
ECCU	(All CCU RAM)	
EMAD	SSAD, SEAD	
SMAD	(All RAM except CCU RAM)	SSAD, SEAD (All RAM except CCU RAM)
COTF	CSAV	CSAV
BRTC	BDIN, BCR1, BCR2, BPPD	
BPAN	BPD1, BPD2, BPD3	
XRTC		XOCW1, XOCW2, XSDW
XSDC	XOCW1, XOCW2, XERR, TCIWC, XCIWP, XCIWC	TCIWC, ECMF, XOCW1, XOCW2, XERR, XCIWC

SORTED BY DATUM

Datum	Value Stored By	Value Fetched By
BCR1	BRTC	ECOM
BCR2	BRTC	ECOM
BDIN	BRTC	EDIS, EMIC, ETIM, ESDW, TPEX, TNXT, PTER, PREQ
BERR		
BERRF		ESDW
PBD1	BPAN	EPP1
PBD2	BPAN	EPP2
PBD3	BPAN	EPP3
BPPD	BRTC	TNCY, PREQ, EPPM

Table B-2. RAM Data Store/Fetch Matrix (Continued)

SORTED BY DATUM (Continued)

Datum	Value Stored By	Value Fetched By
CSAV	COTF	COTF
CTST	TNCY, PTER, PREQ	
ECMF	ECOM	ECOM, XSDC
ED23	EMIC	ESDW, TNCY
EDERR	EDET	
EDETT	EDET	EDET
EDMIT	ETIM, EDET	ETIM, EDET
EDXO	EMIC	EMIC
EDSAV	EADC	
EERR	EGIS, ECOM, ESYN, EDET, EREx	EGIS, ECOM, ESYN, EDET, EREx
EERRF	EGIS, ECOM, ESYN, EDET, EREx	ESDW
EPIE	EPAN, EPPM, EPPI	EPAN, EPPM
EP2E	EPAN, EPPM, EPP2	EPAN, EPPM
EP3E	EPAN, EPPM, EPP3	EPAN, EPPM
EPCD1	EPP1	EPUP
EPCD2	EPP2	EPUP
EPCD3	EPP3	EPUP
EPCF	EPPM	EPPM
EPD1	EPP1	EPP1
EPD2	EPP2	EPP2
EPD3	EPP3	EPP3
EPDT		
EPDVE	EPAN, EPPM, EPP1, EPP2, EPP3	EPAN, EPPM
EPEF1	EREx	EREx
EPEF2	EREx	EREx
EPEF3	EREx	EREx
EPFT	EPAN	EPP1, EPP2, EPP3
EPIR	EPPM	EPPM
EPST	ESYN	ESDW, TSST
EPUF	EPPM, TNCY	TNCY
ER123	EMIC	ESYN, ESDW
ERAM	ERSI	

Table B-2. RAM Data Store/Fetch Matrix (Continued)

SORTED BY DATUM (Continued)

Datum	Value Stored By	Value Fetched By
ERNO	EMIC	EMIC
ERTCF	ERSI, EGIS	EGIS
ESAV	EGIS	EGIS
ESYD	ESYN, EPUP	ESYN
ETCY	ERSI, ETIM, EEOC	ETIM, EEOC, PTER, PREQ
ETST	ETIM	TSST
ISV	EGIS	
PEMAX	EPUP, PREQ	PREQ
PERR	EREx	EREx
PERRF	EREx	ESDW
PFREQ	EEOC, PREQ	EEOC
PIP	PMON, PTER, PREQ	ESDW, TNCY, PMON, PTER, PREQ
PMDT	PMON, PREQ	PMON
PMIF	EEOC, PMON, PREQ	EEOC, PMON, PTER, PREQ
PTAVT	PCIM, PTER	PCIM
PTCP	PREQ	PREQ
PTCS	PREQ	PREQ
PTEI	PREQ	PREQ
PTEM	PCIM, PTER	PCIM, PTER
PTITE	PCIM, PTER	PCIM
PTPMT	PTER	PCIM
SEAD	EMAD, EBKD	SMAD
SSAD	EMAD, EBKD	SMAD
TCDS	TNCY	ESYN, TSST, TNCY, PTER, PREQ
TCIWC	ERSI, TPEX, TNXT, XSDC	XSDC
TCTP	TCOPY, PTER, PREQ	PCIM, PTER, PREQ
TDWL	EPUP	TSST
TEMP		
TEOC		
TERR	TNXT, TNCY	TNXT, TNCY
TERRF	TNXT, TNCY	ESDW
TFST	EEOC, TSST, TNCY, PREQ	TSST
TIN	ERSI, TNXT	TPEX, TNXT, PMON, PCIM, PTER, PREQ

Table B-2. RAM Data Store/Fetch Matrix (Concluded)

SORTED BY DATUM (Continued)

Datum	Value Stored By	Value Fetched By
TINT	ERSI, TPEX, TNXT, PCIM, PTER, PREQ	TPEX, PCIM, PTER, PREQ
TNTP	TCOPY, PTER, PREQ	TCOPY, PCIM, PTER, PREQ
TPM1	ERSI, EPUP	EPAN, TCOPY
TMP2	ERSI, EPUP	EPAN, TCOPY
TPM3	ERSI, EPUP	EPAN, TCOPY
VCBT	PMON, VCLAS	ESDW, PMON
VCDB	VCLAS	
VCDT	PMON	PMON, PTER, PREQ, VCLAS
VCLOG	VCLAS	VCLAS
VCSDB		VCLAS
VCVC	EDET, VCLAS	EDET, VCLAS
XCIWC	XSDC	XSDC
XCIWP	XSDC	
XERR	XSDC	XSDC
XERRF	XSDC	ESDW
XOCW1	XSDC	ECOM, XRTC, XSDC
XOCW2	XSDC	ECOM, XRTC, XSDC
XSDW	EADC, ESDW	EADC, XRTC

Table B-3. Bit Assignments in the Discrete Input Data Word

Bit	Datum	Set (1)	Reset (0)	Source
LSB 0				
1	Reset, R1	R1	Not R1	LMC
2	Reset, R2	R2	Not R2	LMC
3	Reset, R3	R3	Not R3	LMC
4	Dial Select, D2	Select D2	D1 or D3	LMC
5	Dial Select, D3	Select D3	D1 or D2	LMC
6	CMU Conflict	Conflict	Not Conflict	CMU
7	(Unassigned)			
8	(Unassigned)			
9	(Unassigned)			
10	(Unassigned)			
11	(Unassigned)			
12	(Unassigned)			
13	(Unassigned)			
14	Wait/Control	Wait	Control	TCP
MSB 15	Connect/Disconnect	Disconnect	Connect	IFU

Table B-4. Bit Assignments in the Main Street Command Register Status Word

Bit	Signal	Traffic Flow
PBD33J		
LSB 0	Red Ball	NB, SB
1	Yellow Ball	NB, SB
2	Green Ball	NB, SB
3	(Unassigned)	
4	(Unassigned)	
5	(Unassigned)	
6	Don't Walk	NB Ped, SB Ped
7	Flashing Don't Walk	NB Ped, SB Ped
8	Walk	NB Ped, SB Ped
9	(Unassigned)	
10	(Unassigned)	
11	(Unassigned)	
12	(Unassigned)	
13	(Unassigned)	
14	(Unassigned)	
MSB 15	(Update bit, Don't Care)	
PBD14C		
LSB 0	Red Ball	SB
1	Yellow Ball	SB
2	Green Arrow	SB
3	Red Ball	NB
4	Yellow Ball	NB
5	Green Ball	NB
6	Don't Walk	NB Ped, SB Ped
7	Flashing Don't Walk	NB Ped, SB Ped
8	Flashing Walk	NB Ped, SB Ped
9	Red Ball	Bus
10	Yellow Ball	Bus
11	Flashing Yellow Arrow	Bus
12	Green Arrow	Bus
13	Green Arrow	SBLT
14	(Unassigned)	
MSB 15	(Update bit, Don't Care)	

Table B-5. Bit Assignments in the Cross Street Command Register Status Word

Bit	Signal	Traffic Flow
LSB 0	Red Ball	EB, WB
1	Yellow Ball	EB, WB
2	Green Ball	EB, WB
3	(Unassigned)	
4	(Unassigned)	
5	(Unassigned)	
6	Don't Walk	EB Ped, WB Ped
7	Flashing Don't Walk	EB Ped, WB Ped
8	Walk	EB Ped, WB Ped
9	(Unassigned)	
10	(Unassigned)	
11	(Unassigned)	
12	(Unassigned)	
13	(Unassigned)	
14	(Unassigned)	
MSB 15	(Update bit, Don't Care)	

Table B-6. Bit Assignments in the Data Input Word for the Programmable Panel Discrete Switches

Bit	Switch	Set (1)	Reset (0)
LSB 0	Dial 1 Bus Preempt	On	Off
1	Dial 1 Master Controller Synch	On	Off
2	Unassigned		
3	Unassigned		
4	Dial 2 Bus Preempt	On	Off
5	Dial 2 Master Controller Synch	On	Off
6	Unassigned		
7	Unassigned		
8	Dial 3 Bus Preempt	On	Off
9	Dial 3 Master Controller Synch	On	Off
10	Input Request	Yes	No
11	Manual Override	On	Off
12	Unassigned		
13	Unassigned		
13	Unassigned		
14	Unassigned		
MSB 15	Unassigned		

Table B-7. Sequence of Thumbwheel Switch Data in the Data Base Buffers BPD1, BPD2 and BPD3

Displacement	Datum	Units	Number of Units
0	Cycle Length	Seconds	3
1	Offset 1	Seconds	3
2	Offset 2	Seconds	3
3	Offset 3	Seconds	3
4	MSG Split	Percent	2
5	Interval 1	Seconds	3
6	Interval 2 (not used)		

Table B-8. Bit Assignments in the Status Display Output Word

	Bit	Status Condition	Reset (0)	Set (1)
Manual Reset Required	0	Software Disconnect	No	Yes
	1	External Disconnect	No	Yes
	2	CMU Conflict	No	Yes
	3	(Spare)		
	4	Detector Undercount Failure	No	Yes
	5	Other Software-Determined Errors	No	Yes
	6	Last Vehicle Classified	$\overline{\text{Bus}}$	Bus
	7	Preempt in Progress	No	Yes
	8			
	9			
	10	Start Pulse	Off	On
	11	Reset R1	$\overline{\text{R1}}$	R1
	12	Reset R2	$\overline{\text{R2}}$	R2
	13	Reset R3	$\overline{\text{R3}}$	R3
	14	Dial Select D2	D1 or D3	D2
	15	Dial Select D3	D1 or D2	D3

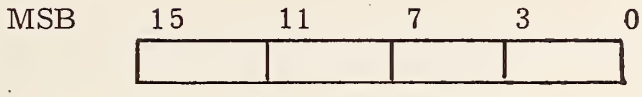
Table B-9. Format of the Main Street Output Command Word (0=Off, 1=On)

Bit	Signal	Traffic Flow
LSB 0	Red Ball	NB, SB
1	Yellow Ball	NB, SB
2	Green Ball	NB, SB
3	(Unassigned)	
4	(Unassigned)	
5	(Unassigned)	
6	Don't Walk	NB Ped, SB Ped
7	Flashing Don't Walk	NB Ped, SB Ped
8	Walk	NB Ped, SB Ped
9	(Unassigned)	
10	(Unassigned)	
11	(Unassigned)	
12	(Unassigned)	
13	(Unassigned)	
14	(Unassigned)	
MSB 15	Update Bit (Set or reset as required)	

Table B-10. Format of the Cross Street Output Command Word (0=Off, 1=On)

Bit	Signal	Traffic Flow
LSB 0	Red Ball	EB, WB
1	Yellow Ball	EB, WB
2	Green Ball	EB, WB
3	(Unassigned)	
4	(Unassigned)	
5	(Unassigned)	
6	Don't Walk	EB Ped, WB Ped
7	Flashing Don't Walk	EB Ped, WB Ped
8	Walk	EB Ped, WB Ped
9	(Unassigned)	
10	(Unassigned)	
11	(Unassigned)	
12	(Unassigned)	
13	(Unassigned)	
14	(Unassigned)	
MSB 15	Update bit (Set or reset as required)	

Table B-11. Format of the Binary-Coded Thumbwheel Data Input Words



Bits 3-0 Units digit

Bits 7-4 Tens digit

Bits 11-8 Hundreds digit (if applicable)

Bits 15-12 (Not Used)

Decimal Digit	4-Bit Binary Code	Hexadecimal Digit
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
(Invalid)	1010	A
(Invalid)	1011	B
(Invalid)	1100	C
(Invalid)	1101	D
(Invalid)	1110	E
(Invalid)	1111	F

APPENDIX C

BUS PREEMPT CONTROL UNIT PROGRAM FOR WASHINGTON INSTALLATION

3.0 PROGRAM DOCUMENTATION

Due to the high degree of commonality between this field test program and PBD33J, only the differences from that program are included herein. This document is to be treated as an addendum to the PBD33J document in Appendix B.

This document is written according to the FHWA documentation standard.

3.0.1 Identification

This document describes the computer ptoram identified as the Passive Bus Detector Software for the field test at Fourteenth and C Streets in Washington, D.C. This program is also identified by the acronym, PBD14C. This software is operational in the microcomputer unit (MCU) of the Bus Preempt Control Unit (BPCU) for the demonstration of the concept of bus classification and preemption of traffic signal timing at the field test site.

3.0.2 Functional Summary

This program generates the control commands to the traffic signal display, incorporating the preemption of the timing plan for clearance of a standard transit bus that is detected and classified. In the absence of preemption, the traffic signals are controlled according to the current background fixed timing plan in synchronization with the loop master controller.

Data for the three timing plans, preemption extended green time and synchronization with the loop master controller, are input to this program from the programming panel on the front of the BPCU.

This program operates in a real time, priority interrupt environment in the MCU and performs all the functions of vehicle signature classification, preemption, traffic signal timing, executive monitoring, scheduling and self testing. The program is written in the assembly language of the MCU.

Nomenclature

The definitions of abbreviations used in this document are listed in Table B-1.

Memory Map

See Figure B-1.

Memory Size of Program Components

Table B-2 contains the memory size of the program components with the following modifications:

Routine	Size (Hexadecimal Words)
Timing Plan Tables	95
PMON	97
PCIM	0
PTER	0
PREQ	0

PBUF contains the following data:

PIP - Preempt in Progress Flag. Set by Preempt Request Routine (PREQ) when a preempt condition is initialized. PIP is used to determine if a preempt is in progress.

PTEM - Preempt Temporary Storage Word. Used for short-term storage of intermediate results.

PTCP - Primary Contractible Internal Address. Not used.

PTCS - Secondary Contractible Interval Address. Not used.

PTEI - Primary Extendible Interval Address. Not used.

PFREQ - Preempt Request Flag. PFREQ is set by the Preempt Request Routine (PREQ) following the modification of the timing plan. When PFREQ is non-zero, the End of Cycle Routine (EEOC) resets the "wait for synchronized start" flag and thus prevents the wait for the synchronized start dwell from disrupting the preempt operation.

PTAVT - Available Time for Interval Restoration. Not used.

PTITE - Interval to be Restored. Not used.

PTPMT - Address of First Location in Master Timing Plan. Not used.

PEMAX - Maximum Preempt Extension Parameters. PEMAX is a four-word array that contains:

PEMAX(0) - The maximum extension allowed for the current dial setting.

PEMAX(1) - The maximum extension allowed for the first dial setting.

PEMAX(2) - The maximum extension allowed for the second dial setting.

PEMAX(3) - The maximum extension allowed for the third dial setting.

PEMAX(0) - Is set by Preempt Request Routine (PREQ), which reads the current dial setting (TCDS) and copies the proper word. PREQ uses PEMAX(0) to limit expansion of the extendible interval (EI).

VCDT - Vehicle Detector Index. Used as a communication link between PMON, PREQ, PTER, and VCLAS. VCDT identifies the detector currently under scrutiny and hence the approach.

VCBT - Recent Bus Classification Timer. Set by the Vehicle Classification Routine, VCLAS, whenever a vehicle is classified as a bus. A non-zero value in VCBT causes a 1 to be set in bit 6 of the status display word. A positive value in VCBT is decrement by 1 in the Preempt Monitor Routine (PMON).

VCVC - Vehicle Counters. An array 32 words long, each word incremented when the corresponding vehicle detector records a vehicle signature. Used in vehicle detector failure monitoring.

VCDB - Vehicle Classifier Data Base. An array 25 words long. The following is copied from the Signature Data Block (SDB):

VCDB(1) Total duration of signature (x 4 msec).

VCDB(2) First peak amplitude.

VCDB(3) Time from "turn on" to first peak (x 4 msec).

VCDB(4) Second peak amplitude.

VCDB(5) Time from "turn on" to second peak (x 4 msec).

VCDB(6) Last peak amplitude.

VCDB(7) Time from "turn on" to last peak (x 4 msec).

The following is determined during the moving/stopped test.

VCDB(8) (VCDB(1) + (VCDB(3) - (VCDB(7))

The following is determined during the stopped vehicle classifier.

VCDB(9) (G1 coefficient for H1 function) * (VCDB(4))

VCDB(10) (G2 coefficient for H1 function)*(VCDB(6))

VCDB(11) (G1 coefficient for H2 function)*(VCDB(4))

VCDB(12) (G2 coefficient for H2 function)*(VCDB(6))

VCDB(13) H1 function

VCDB(14) H2 function

The following are determined in the moving vehicle classifier.

VCDB(15) (G1 coefficient for K4/K5 function) * (VCDB(4))

VCDB(16) (G2 coefficient for K4/K5 function) * (VCDB(6))

VCDB(17) (VCDB(16) - (VCDB(17)/(VCDB(2))

VCDB(18) (F1 coefficient for K4/K5 function) * (VCDB(5) - (VCDB(5))

VCDB(19) (F2 coefficient for K4/K5 function) * (VCDB(7) - (VCDB(5))

VCDB(20) (VCDB(19) - (VCDB(18)/(VCDB(8)))

VCDB(21) K4 function

VCDB(22) K5 function

VCDB(23-25) Not used.

VCLOG - Vehicle Classifier Vehicle Log. An array 15 words long.

VCLOG(1) - Number of signatures rejected as noise (total duration less than 128 msec).

VCLOG(2) - Number of signatures that failed the gross features test and are thus classified non-buses (fewer than three peaks).

VCLOG(3) - Number of signatures classified by stopped vehicle classifier.

VCLOG(4) - Number of signatures classified by moving vehicle classifier.

VCLOG(5) - Number of signatures that passed gross features test and were classified as non-buses.

VCLOG(6) - Number of signatures classified as buses.

VCLOG(7-15) - Not used.

3.6.6 TIMING PLAN DATA TABLES

"Timing plan data tables" refers to the ROM storage area for fixed timing plan parameters. Definitions of these parameters follow.

TSCI - This is the index of the synchronization contractible interval. TSCI is not used in the current version of the PBD software. It may be used in later applications in which the timing plan CPC has more control of synchronization.

THI - This is the index of the transfer hold interval.

TAI1 - This is the index of the first adjustable interval in the timing plan.

TAI2 - This is the index of the second adjustable interval in the timing plan.

TIMX - This is the maximum value allowed for the interval index (TIN). If TIN exceeds TIMX during exeuction of the TNXT routine, an error is recorded and a jump to the auto-disconnect routine (EADC) is executed.

TECI - This is the beginning location of a table containing contractible, extendible, and exchange interval information. The locations are allocated as follows:

TECI+1 First interval in extension request range.

TECI+2 Extendible interval index.

TECI+3 Incremental extension time.

TECI+4 Primary contractible interval index.

TECI+5 Secondary contractible interval index.

TECI+6 Tertiary contractible interval index.

TECI+7 Number of interval pairs to be exchanged.

TECI+8 Indices of exchange pair 1

TECI+9 Indices of exchange pair 2

The first 8 bits of an exchange pair index word specify the non-bus interval whose duration is to be transferred

to the bus interval. The non-bus interval duration is then set to zero. The bus interval is specified by the last 8 bits of the exchange pair index word.

TCIW - This is the beginning location of the buffer for the command index word table. The buffer presently contains 20 locations, and for a timing plan with n intervals would be allocated as follows:

Relative Location	Contents
1	Command index word for first timing interval
2	Command index word for second timing interval
.	.
.	.
.	.
.	.
.	.
.	.
.	.
n	Command index word for nth timing interval

Any additional locations in the buffer are blank.

TPTAB - This is the beginning location of the three skeleton master timing plan tables. There are 60 locations, the last of which is labeled TPE. The first 20 locations contain the dial 1 skeleton master timing plan; the next 20 locations contain the dial 2 master; and the third 20 locations contain the dial 3 master. All 60 locations are copied into the read/write data base tables at initialization, dial 1 into TPM1, dial 2 into TPM2, and dial 3 into TMP3.

TPE - This is the last location in the skeleton master timing plan tables. Its address is used as a signal to terminate the copy routine during initialization.

3.7 BUS PREEMPT CPC

3.7.1 Preempt Monitor Routine (PMON)

3.7.1.1 Program Narrative

3.7.1.1.1 Purpose

This routine monitors preempt operation for initiating preempt modifications of the timing plan.

3.7.1.1.2 Software Interfaces

The preempt monitor routine (PMON) is called only by the executive scheduling routine, ESCH, during each real time clock interval. PMON calls VCLAS to examine a particular signature in the data base. The last datum of the signature is the end of signature status. Only completed signatures are classified. For classification of a signature as bus, VCLAS returns to PMON with AC0 set to zero, otherwise AC0 is set to one.

3.7.1.1.3 Assumptions

There is only one detector on the preemptive approach to the intersection. This detector functions as the preempt request classification detector. There is no preempt clearance classification detector.

3.7.1.1.4 Solution Approach

A preempt in progress flag is used to indicate a preempt modification of the timing plan due to an upstream classification of a bus. Preempt modification consists of interval replacement in the next cycle of the timing plan and interval extension during timing of the extendible interval. A preempt in progress allows extension modifications during the extendible interval but precludes the interval replacement operation and synchronization dwell, dial change, and programming panel update. The preempt is terminated when the extendible interval is timed out.

The timing plan tables are structured to include both the preempt and non-preempt interval sequences. The master timing plans contain zero duration preempt intervals. At preempt initiation, the interval durations are transferred from the non-preempt sequence to the preempt sequence in the next cycle timing plan. Subsequently, the extendible interval in the preempt sequence may be extended.

The preempt is initiated by a bus classification during that sequence of the timing plan for which the vehicles on the preempt approach may not clear the intersection. Thus the preempt sequence appears in the next cycle for those vehicles that could not clear during this cycle.

3.7.1.1.5 Program Operation

The program begins by decrementing a positive recent bus classification timer by one. The interval index is tested to determine when the preempt sequence has passed such that the preempt in progress and preempt modification flags can be reset.

The vehicle classifier is called to examine each signature data block for a completed signature. The classification result for a completed signature is returned in AC0. Following classification of a bus, this routine determines if preempt modification or preempt extension is warranted. Preempt modification of the next cycle timing plan depends on when the classification occurs in the current timing plan. Because preempt extension can be made only during timing of the preempt extension interval, it involves updating the current interval timer, TINT, as well as the timing plan table.

3.7.1.1.6 Model

Not applicable.

3.7.1.1.7 Date

August 29, 1975

3.7.1.1.8 Remarks

None

3.7.1.2 Program Work Area

Refer to data base section for complete definitions.

3.7.1.2.1 Switches

BPPD Panel preempt on/off switches.

3.7.1.2.2 Counters and Accumulators

PIP - Preempt in Progress Flag

PFREQ - Preempt Request Flag

TIN - Current interval index

TINT - Current interval timer

VCBT - Recent bus classification timer

Hold Areas

ATCTP - Address pointer to current cycle timing plan table

ATNTP - Address pointer to next cycle timing plan table

PTEM - Temporary storage word

TCDS - Current dial setting

TFST - Wait for synch start flag

TECI - Approach parameters

TECI+1 - First interval in extension request range

TECI+2 - Extendible interval index

TECI+3 - Incremental extension time

TECI+4 - Primary contractible interval index

TECI+5 - Secondary contractible interval index

TECI+6 - Tertiary contractible interval index

TECI+7 - Number of interval pairs to be exchanged

TECI+8 - Indices of exchange pair 1

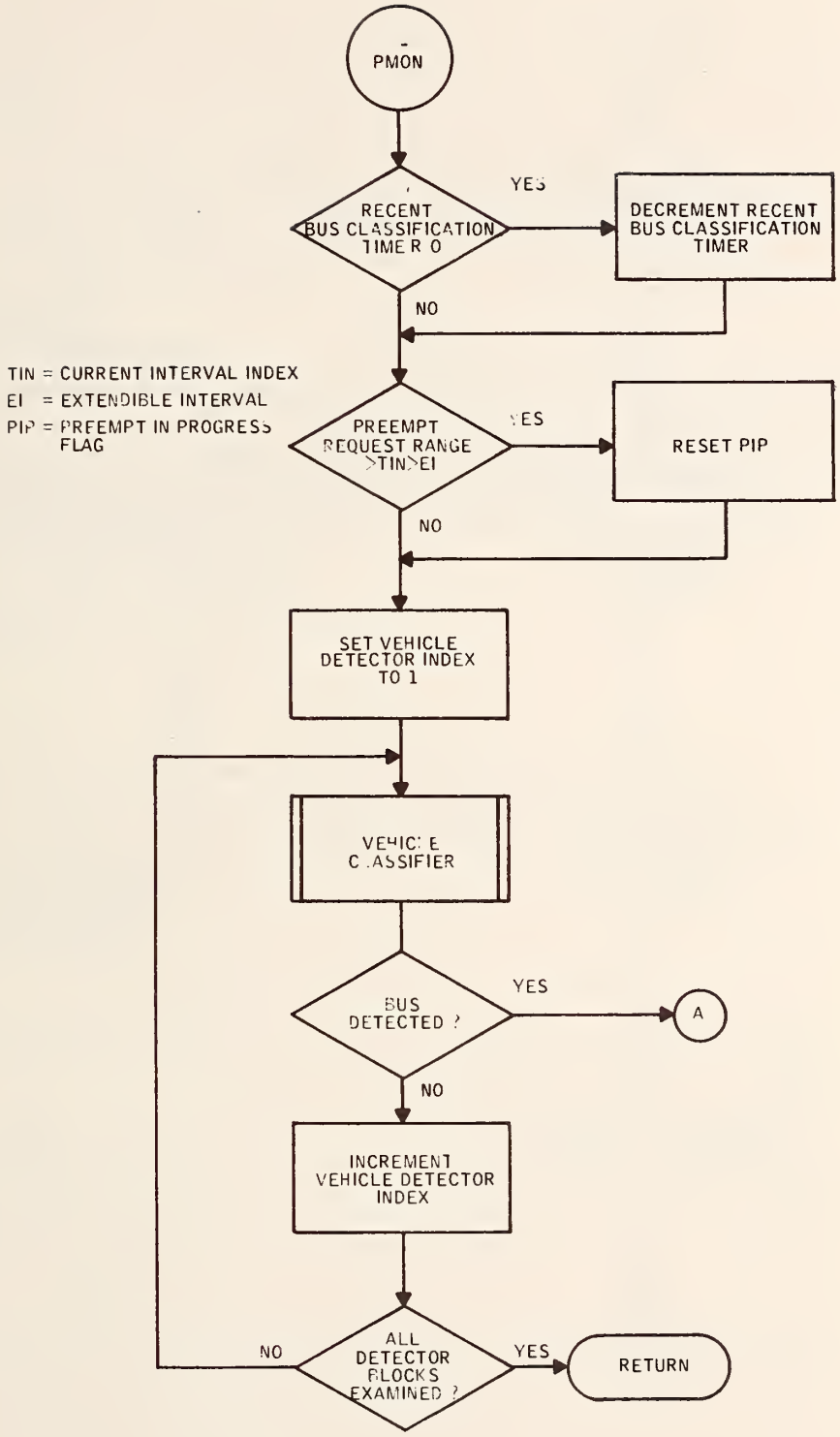
TECI+9 - Indices of exchange pair 2. The first 8 bits of an exchange pair index word specifies the non-bus interval whose duration is to be transferred to the bus interval. The non-bus interval duration is then set to zero. The bus interval is specified by the last 8 bits of the exchange pair index word.

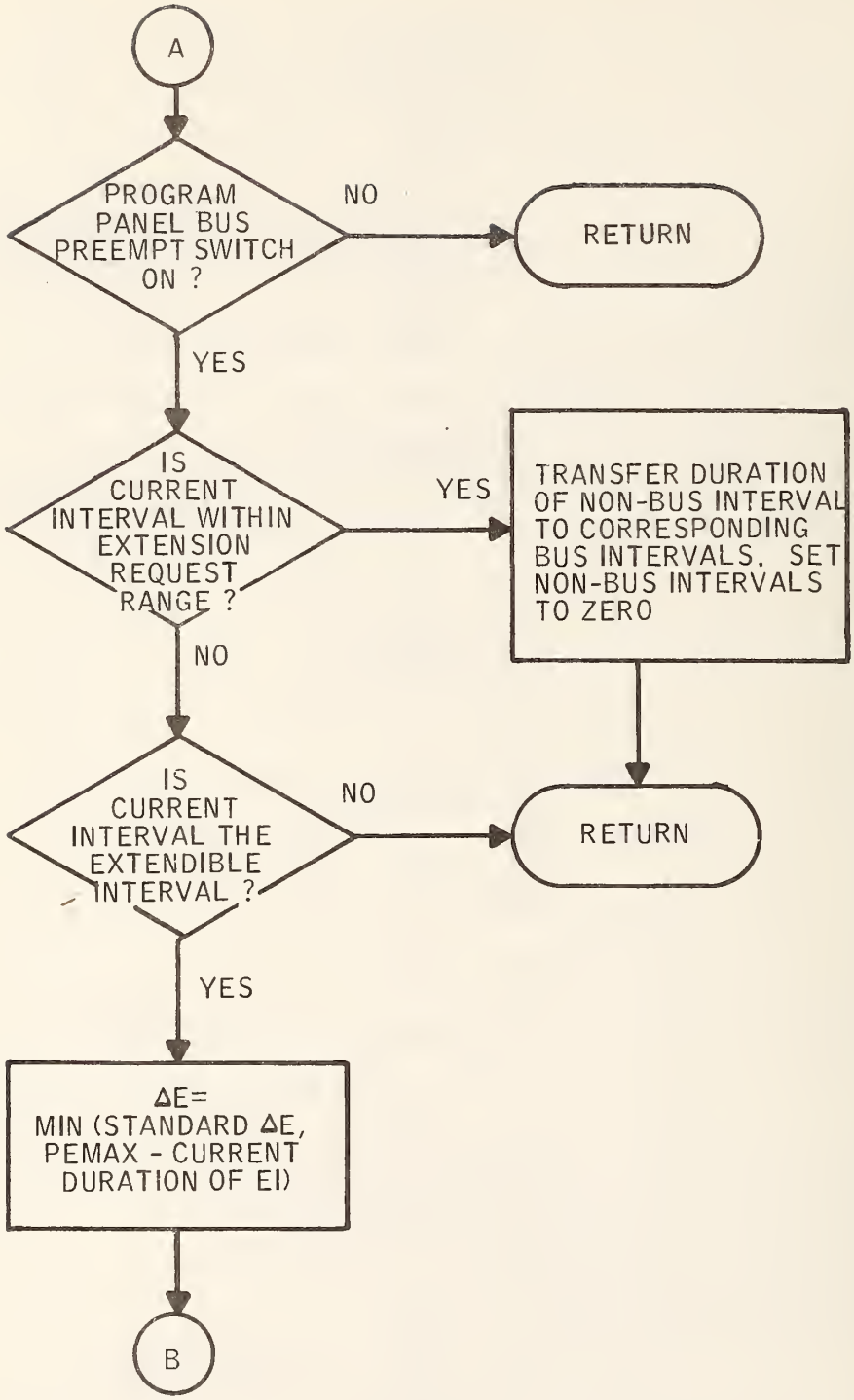
Local Data

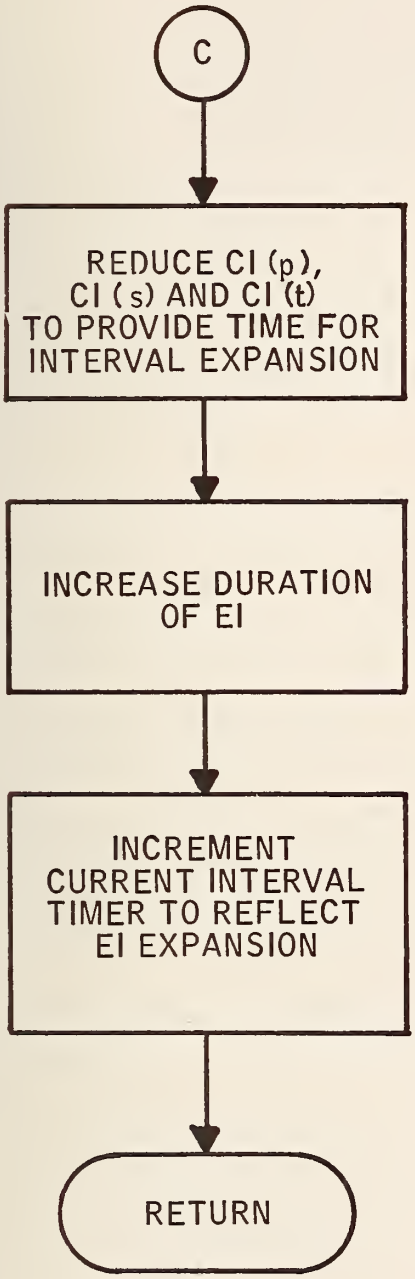
PMAS - Masks used to determine the state of the control panel preempt on/off switch for the current dial settings.

3.7.1.3 Program Flowchart

The program flowchart is presented in Figure C-1.


PMON
RECENT BUS CLASSIFICATION TIME R 0
YES
DECREMENT RECENT BUS CLASSIFICATION TIMER
NO
TIN = CURRENT INTERVAL INDEX
EI = EXTENDIBLE INTERVAL
PIP = PREEMPT IN PROGRESS FLAG
PREEMPT REQUEST RANGE >TIN>EI
YES
RESET PIP
NO
SET VEHICLE DETECTOR INDEX TO 1
VEHICLE CLASSIFIER
BUS DETECTED ?
YES
A
NO
INCREMENT VEHICLE DETECTOR INDEX
ALL DETECTOR BLOCKS EXAMINED ?
NO
YES
RETURN


A
PROGRAM PANEL BUS PREEMPT SWITCH ON ?
NO
RETURN
YES
IS CURRENT INTERVAL WITHIN EXTENSION REQUEST RANGE ?
YES
TRANSFER DURATION OF NON-BUS INTERVAL TO CORRESPONDING BUS INTERVALS. SET NON-BUS INTERVALS TO ZERO
NO
IS CURRENT INTERVAL THE EXTENDIBLE INTERVAL ?
NO
RETURN
YES
ΔE= MIN (STANDARD ΔE, PEMAX - CURRENT DURATION OF EI)
B



CI (p) = PRIMARY CONTRACTIBLE INTERVAL
CI (s) = SECONDARY CONTRACTIBLE INTERVAL
CI (t) = TERTIARY CONTRACTIBLE INTERVAL

Figure C-1. Preempt Monitor - PDB14C (Concluded)

3.7.1.4 Program Listing

The program listing is presented in Appendix D.

3.7.1.5 Input Data Formats

Not applicable.

3.7.1.6 Report Formats

Not applicable.

3.7.1.7 Test Data

To be determined.

3.7.1.8 Revisions

None.

3.8 PROGRAM LISTINGS AND FLOWCHARTS

The complete program listing, including the assembly constants and the cross reference table of programming symbols, is presented in Appendix D, and the flowcharts are included with the program documentation of each routine.

GPO 907-805

TE 662 U
.A3
no.FHWA-RD-
76-67 BORROWER

DOT LIBRARY
00055534